

TUGAS AKHIR - KI141502

RANCANG BANGUN APLIKASI EDUKASI INTERAKTIF UNTUK MELATIH BERPIKIR KOMPUTASIONAL MATERI COMBINATORIAL

CAYZA ANGKA MAULANA
NRP 5113100121

Dosen Pembimbing
Rully Soelaiman, S.Kom., M.Kom.
Fajar Baskoro, S.Kom., M.T.

JURUSAN TEKNIK INFORMATIKA
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Surabaya 2017

[Halaman ini sengaja dikosongkan]



TUGAS AKHIR - KI141502

RANCANG BANGUN APLIKASI EDUKASI INTERAKTIF UNTUK MELATIH BERPIKIR KOMPUTASIONAL MATERI COMBINATORIAL

**CAYZA ANGKA MAULANA
NRP 5113100121**

**Dosen Pembimbing
Rully Soelaiman, S.Kom., M.Kom.
Fajar Baskoro, S.Kom., M.T.**

**JURUSAN TEKNIK INFORMATIKA
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Surabaya 2017**

[Halaman Ini Sengaja Dikosongkan]



FINAL PROJECT- KI141502

DESIGN OF INTERACTIVE EDUCATIONAL APPLICATION FOR LEARNING COMPUTATIONAL THINKING COMBINATORICS MODULE

**CAYZA ANGKA MAULANA
NRP 5113100121**

**Advisor
Rully Soelaiman, S.Kom., M.Kom.
Fajar Baskoro, S.Kom., M.T.**

**DEPARTMENT OF INFORMATICS
Faculty of Information Technology
Sepuluh Nopember Institute of Technology
Surabaya 2017**

[Halaman ini sengaja dikosongkan]

LEMBAR PENGESAHAN

RANCANG BANGUN APLIKASI EDUKASI INTERAKTIF UNTUK MELATIH BERPIKIR KOMPUTASIONAL MATERI COMBINATORIAL

TUGAS AKHIR

**Diajukan Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada**

**Rumpun Mata Kuliah Algoritma dan Pemrograman
Program Studi S-1 Jurusan Teknik Informatika
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember**

Oleh:

**CAYZA ANGKA MAULANA
NRP. 5113 100 121**

Disetujui oleh Dosen Pembimbing Tugas Akhir:

Rully Soelaiman, S.Kom., M.Kom.

NIP: 197002131994021001

Fajar Baskoro, S.Kom., M.T.

NIP: 197404031999031002



**SURABAYA
JULI, 2017**

[Halaman ini sengaja dikosongkan]

RANCANG BANGUN APLIKASI EDUKASI INTERAKTIF UNTUK MELATIH BERPIKIR KOMPUTASIONAL MATERI *COMBINATORIAL*

Nama Mahasiswa : Cayza Angka Maulana
NRP : 5113100121
Jurusan : Teknik Informatika FTIf-ITS
Dosen Pembimbing I : Rully Soelaiman, S.Kom., M.Kom.
Dosen Pembimbing II : Fajar Baskoro, S.Kom., M.T.

ABSTRAK

Berpikir komputasional (computational thinking) adalah metode menyelesaikan persoalan dengan menerapkan teknik ilmu komputer (informatika). Berpikir Komputasional memiliki peran penting dalam pengembangan aplikasi komputer, namun berpikir komputasional juga dapat digunakan untuk mendukung pemecahan masalah di semua disiplin ilmu, seperti humaniora, matematika, dan ilmu pengetahuan. Seperti bermain musik dan belajar bahasa asing, berpikir komputasional melatih otak untuk terbiasa berfikir secara logis, terstruktur, dan kreatif.

Bebras merupakan aktivitas ekstra kurikuler yang mengedukasi kemampuan problem solving dalam informatika dengan jumlah peserta terbanyak di dunia. Yang dilombakan dalam kompetisi ini adalah sekumpulan soal yang disebut Bebras task. Bebras task disajikan dalam bentuk uraian persoalan yang dilengkapi dengan gambar yang menarik, sehingga siswa dapat lebih mudah memahami soal. Soal-soal tersebut dapat dijawab tanpa perlu belajar informatika terlebih dahulu, tapi soal tersebut sebetulnya terkait pada konsep tertentu dalam informatika dan computational thinking. Soal-soal pada tugas akhir ini dikembangkan pada perangkat bergerak serta web supaya pengguna dapat mengakses aplikasi ini dengan mudah. Materi soal pada aplikasi ini adalah combinatorial. Pada aplikasi ini, pengguna diharuskan untuk login dan register. Setelah itu,

pengguna dapat mengerjakan soal-soal yang diberikan pada aplikasi dan melihat hasil yang didapat oleh pengguna.

Pengujian aplikasi dilakukan dengan menggunakan perangkat bergerak dan juga melalui kuesioner responden yang berminat untuk mengerjakan soal-soal yang disediakan oleh aplikasi. Berdasarkan hasil kuesioner responden dapat diketahui tingkat kenyamanan antarmuka mendapatkan nilai 3.9 dari nilai maksimal 5, performa aplikasi mendapatkan nilai 4.1 dari nilai maksimal 5, dan materi soal mendapatkan nilai 4.6 dari nilai maksimal 5. Dengan dikembangkannya aplikasi ini, diharapkan dapat menjadi alternatif latihan berpikir komputasional oleh pengguna.

Kata kunci: Berpikir Komputasional, Bebras, Bebras Task, Kompetisi Bebras.

DESIGN OF INTERACTIVE EDUCATIONAL APPLICATION FOR LEARNING COMPUTATIONAL THINKING COMBINATORICS MODULE

Student Name : Cayza Angka Maulana
NRP : 5113100121
Major : Informatics Engineering FTIf-ITS
Advisor I : Rully Soelaiman, S.Kom., M.Kom.
Advisor II : Fajar Baskoro, S.Kom., M.T.

ABSTRACT

Computational thinking is a method to solve problems by implements computer science technique. Computational thinking has an important role in development computer application, but computational thinking can also be used to help solving problems almost at all materials, such as humanities, mathematics, and science. Just like playing music or learning foreign languages, computational thinking train our brain to get used to think logically, structured, and creative.

Bebras is an extracurricular activity that improves problem solving ability in informatics with the most participant in the world. At the same time, Bebras Task shows aspects of informatics and computational thinking. Bebras Competition is held every year. The main purpose of this challenge is to promote informatics and computational thinking to teachers and young people in particular. Bebras task is presented with an attractive picture, so that students can more easily understand the problem. The tasks can be answered without studying computer science first. The tasks in this final year project developed on mobile and web so that users can access this application easily. In this application, users must register and login before they can answer the questions on this application. After answer all questions from application, user can see their score.

Testing conducted using mobile device and using questionnaire on respondents who are interested to do the tasks provided by the application. Based on respondent questionnaire, the comfort level of interface get score 3.9 from maximum score 5, application performance get score 4.1 from maximum score 5, and material task get score 4.6 from maximum score 5. By the development of this application, is expected to be an alternative exercise to learn computational thinking.

Key words: *Computational Thinking, Bebras, Bebras Task, Bebras Competition.*

KATA PENGANTAR

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Segala puji dan syukur senantiasa penulis panjatkan ke hadirat Allah Subhanahu Wa Ta'ala yang telah melimpahkan rahmat dan hidayah-Nya sehingga penulis masih dikaruniai kenikmatan Islam dan iman. Sholawat serta salam tak lupa tercurah kepada Insan pilihan, Khotimul Anbiya' wal Mursalin, Nabi Muhammad Shollallahu 'Alaihi Wassalam yang dinantikan safaatnya kelak.

Alhamdulillah wa syukurlillah pada kesempatan kali ini penulis dapat menyelesaikan tugas akhir yang berjudul “Rancang Bangun Aplikasi Edukasi Interaktif untuk Melatih Berpikir Komputasional Materi Combinatorial”. Pengerjaan tugas akhir ini adalah saat bagi penulis untuk menerapkan ilmu-ilmu yang penulis dapatkan semenjak awal perkuliahan sampai hari ini di lingkungan jurusan Teknik Informatika Institut Teknologi Sepuluh Nopember Surabaya

Dalam pelaksanaan dan pembuatan tugas akhir ini, penulis menerima banyak bantuan dari berbagai pihak, oleh karena itu melalui lembar kata pengantar ini, penulis menyampaikan ucapan terima kasih kepada:

1. Kedua orang tua penulis, yang selalu mencurahkan doa, dukungan semangat, perhatian, serta kasih sayang kepada penulis.
2. Kakak-kakak dan saudara penulis yang selalu memberi dukungan selama menempuh masa kuliah.
3. Bapak Rully Soelaiman dan Bapak Fajar Baskoro selaku dosen pembimbing yang telah bersedia meluangkan waktu untuk memberikan petunjuk selama pengerjaan Tugas Akhir.
4. Sahabat-sahabati Sepuluh Nopember yang menemani penulis dalam hal semangat spiritual.

5. Para admin laboratorium IGS (Interaksi Grafika dan Seni) beserta IGSahabat sebagai tempat tukar ilmu dan pengalaman selama pengerjaan tugas akhir.
6. Juga tidak lupa kepada semua pihak yang belum sempat disebutkan satu per satu yang telah membantu terselesaikannya tugas akhir ini, penulis mengucapkan terima kasih.

Penulis telah mengerjakan sebaik mungkin dalam penyusunan aplikasi tugas akhir ini. Penulis juga memohon maaf apabila terdapat kekurangan, kesalahan, maupun kelalaian yang telah penulis lakukan.

Surabaya, Januari 2017

Cayza Angka Maulana

DAFTAR ISI

LEMBAR PENGESAHAN.....	v
ABSTRAK	vii
ABSTRACT	ix
KATA PENGANTAR.....	xi
DAFTAR ISI	xiii
DAFTAR GAMBAR	xvii
DAFTAR TABEL	xxi
DAFTAR KODE SUMBER	xxiii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah	2
1.4 Tujuan.....	2
1.5 Manfaat.....	2
1.6 Metodologi	3
1.7 Sistematika Penulisan	4
BAB II TINJAUAN PUSTAKA.....	7
2.1 Soal Bebras.....	7
2.2 WebGL	7
2.3 Combinatorics.....	8
2.4 Unity	8
2.5 Pemrograman C#	9
2.6 Visual Studio	10
2.7 Struktur data stack dan queue.....	11
2.8 Aplikasi Terkait yang Sudah Ada.....	12
BAB III ANALISIS DAN PERANCANGAN	13
3.1 Analisis Sistem	13
3.1.1 Analisis Permasalahan	13
3.1.2 Deskripsi Umum Sistem	14
3.1.3 Spesifikasi Kebutuhan Perangkat Lunak	14
3.2 Perancangan.....	32
3.2.1 Lingkungan Perancangan Perangkat Lunak.....	32

3.2.2	Perancangan Arsitektur Sistem	33
3.2.3	Perancangan Struktur Data	34
3.2.4	Perancangan Kelas Diagram	35
3.2.5	Perancangan Antarmuka Pengguna	36
	BAB IV IMPLEMENTASI.....	49
4.1	Lingkungan Implementasi	49
4.2	Implementasi Antarmuka Pengguna	50
4.2.1	Implementasi Antarmuka Halaman Register Pengguna	50
4.2.2	Implementasi Antarmuka Halaman Login Pengguna	55
4.2.3	Implementasi Antarmuka Halaman Beranda	60
4.2.4	Implementasi Antarmuka Halaman Kompetisi	60
4.2.5	Implementasi Antarmuka Halaman Detail Peserta	61
4.2.6	Implementasi Halaman Antarmuka Tentang Bebras	64
4.2.7	Implementasi Antarmuka Halaman Penjelasan Soal	65
4.2.8	Implementasi Antarmuka Halaman Selesai	65
4.2.9	Implementasi Antarmuka Halaman Ranking	69
4.3	Implementasi Pembuatan Aplikasi	71
4.3.1	Implementasi Pembuatan Scene	71
4.3.2	Implementasi Import Asset	71
4.3.3	Implementasi Load Objek ke Scene	72
4.3.4	Implementasi Pembuatan Script	73
4.3.5	Implementasi Pembuatan Soal	74
4.3.6	Implementasi Random Soal dan Insert Nilai	81
4.3.7	Implementasi Waktu Hitung Mundur Soal	85
4.3.8	Implementasi Terjemahan Soal	88
	BAB V PENGUJIAN DAN EVALUASI	93
5.1	Lingkungan Uji Coba	93
5.2	Pengujian Fungsionalitas	93
5.2.1	Pengujian Fungsionalitas Melihat Tentang Bebras	94
5.2.2	Pengujian Fungsionalitas Melihat Detail Peserta	96
5.2.3	Pengujian Fungsionalitas Mengerjakan Soal	98
5.2.4	Pengujian Fungsionalitas Melihat Nilai	100
5.3	Evaluasi Pengujian	103

5.3.1	Evaluasi Pengujian Fungsionalitas	103
5.3.2	Evaluasi Pengujian Aplikasi Terhadap Pengguna ..	104
	BAB VI KESIMPULAN DAN SARAN	107
6.1.	Kesimpulan.....	107
6.2.	Saran.....	107
	DAFTAR PUSTAKA	109
	LAMPIRAN A PANDUAN PENGGUNAAN APLIKASI..	
	111
	LAMPIRAN B PANDUAN PEMBUATAN SOAL	123
	LAMPIRAN C PANDUAN MENJAWAB SOAL.....	135
	LAMPIRAN D KUISIONER	143

DAFTAR GAMBAR

Gambar 2.1 Tampilan Antarmuka Visual Studio	11
Gambar 2.2 Struktur data stack	11
Gambar 2.3 Tampilan Antarmuka lightbot	12
Gambar 3.1 DFD Level 0	15
Gambar 3.2 DFD Level 1 untuk Proses Registrasi.....	16
Gambar 3.3 DFD Level 1	17
Gambar 3.4 Diagram Kasus Penggunaan.....	19
Gambar 3.5 Diagram Aktifitas Melihat Tentang Bebras.....	21
Gambar 3.6 Diagram Aktifitas Melihat Detail Peserta.....	22
Gambar 3.7 Diagram Sekuens Melihat Detail Peserta	23
Gambar 3.8 Robustness Diagram Melihat Detail Peserta	23
Gambar 3.9 Diagram Aktifitas Mengerjakan Soal	26
Gambar 3.10 Diagram Sekuens Mengerjakan Soal.....	27
Gambar 3.11 Robustness Diagram Mengerjakan Soal.....	27
Gambar 3.12 Diagram Aktifitas Melihat Nilai.....	28
Gambar 3.13 Diagram Sekuens Melihat Nilai	29
Gambar 3.14 Robustness Diagram Melihat Nilai.....	29
Gambar 3.15 Diagram Aktifitas Melihat Ranking	31
Gambar 3.16 Diagram Sekuens Melihat Ranking	31
Gambar 3.17 Robustness Diagram Melihat Ranking	32
Gambar 3.18 Arsitektur Sistem	33
Gambar 3.19 Conceptual Data Model.....	35
Gambar 3.20 Physical Data Modeling.....	35
Gambar 3.21 Diagram Kelas Aplikasi.....	36
Gambar 3.22 Rancangan Antarmuka Register	37
Gambar 3.23 Rancangan Antarmuka Login.....	38
Gambar 3.24 Rancangan Antarmuka Beranda	39
Gambar 3.25 Rancangan Antarmuka Kompetisi	40
Gambar 3.26 Rancangan Antarmuka Detail Peserta	41
Gambar 3.27 Rancangan Antarmuka Tentang Bebras	42
Gambar 3.28 Rancangan Antarmuka Penjelasan Soal	43
Gambar 3.29 Rancangan Antarmuka Soal Pertama	44

Gambar 3.30 Rancangan Antarmuka Soal Kedua	46
Gambar 3.31 Rancangan Antarmuka Selesai	47
Gambar 3.32 Rancangan Antarmuka Ranking	48
Gambar 4.1 Implementasi Antarmuka Halaman Register Pengguna	50
Gambar 4.2 Implementasi Antarmuka Halaman Login Pengguna	56
Gambar 4.3 Implementasi Halaman Antarmuka Beranda	60
Gambar 4.4 Implementasi Halaman Antarmuka Kompetisi.....	60
Gambar 4.5 Implementasi Halaman Antarmuka Detail Peserta..	61
Gambar 4.6 Implementasi Halaman Antarmuka Tentang Bebras	64
Gambar 4.7 Implementasi Antarmuka Halaman Ranking.....	69
Gambar 4.8 Pembuatan Scene	71
Gambar 4.9 Implementasi Import Asset.....	72
Gambar 4.10 Implementasi Load Objek ke Scene	73
Gambar 4.11 Implementasi Pembuatan Script	74
Gambar 5.1 Halaman Beranda.....	95
Gambar 5.2 Halaman Tentang Bebras.....	96
Gambar 5.3 Halaman Beranda.....	97
Gambar 5.4 Halaman Detail Peserta.....	97
Gambar 5.5 Halaman Penjelasan Soal.....	98
Gambar 5.6 Halaman Soal.....	99
Gambar 5.7 Halaman Soal.....	100
Gambar 5.8 Halaman Selesai.....	101
Gambar 5.9 Halaman Selesai.....	102
Gambar 5.10 Halaman Ranking	103
Gambar A.1 Halaman Beranda.....	111
Gambar A.2 Halaman Detail Peserta.....	112
Gambar A.3 Halaman Tentang Bebras.....	113
Gambar A.4 Halaman Kompetisi	114
Gambar A.5 Halaman Login	115
Gambar A.6 Halaman Register.....	116
Gambar A.7 Halaman Penjelasan Soal.....	117
Gambar A.8 Halaman Soal.....	118

Gambar A.9 Halaman Selesai	120
Gambar A.10 Halaman Ranking	121
Gambar B.1 Tampilan Awal Unity	123
Gambar B.2 Tampilan Awal	124
Gambar B.3 Ubah Perspective Kamera.....	124
Gambar B.4 Membuat Game Object.....	125
Gambar B.5 Menambah Sprite Renderer untuk Game Object ..	125
Gambar B.6 Memilih UISprite untuk Sprite Renderer.....	126
Gambar B.7 Menduplikasi Game Object	126
Gambar B.8 Mewarnai Game Object	127
Gambar B.9 Mengatur Letak Game Object.....	127
Gambar B.10 Membuat Canvas	129
Gambar B.11 Memilih Warna Canvas	129
Gambar B.12 Konfirmasi Warna Canvas	130
Gambar B.13 Membuat Teks	130
Gambar B.14 Membuat Tombol	131
Gambar B.15 Membuat Script.....	132
Gambar B.16 Drag Script menuju Object	132
Gambar B.17 Reference Object pada Script.....	133
Gambar C.1 State Awal.....	135
Gambar C.2 State Akhir	136
Gambar C.3 Langkah Pertama	137
Gambar C.4 Langkah Kedua	137
Gambar C.5 Langkah Ketiga dan Keempat.....	138
Gambar C.6 Langkah Kelima dan Keenam.....	139
Gambar C.7 State Awal Soal Kedua	140
Gambar C.8 State Akhir Soal Kedua.....	140
Gambar D.1 Kuisioner Responden Pertama.....	143
Gambar D.2 Kuisioner Responden Kedua	144
Gambar D.3 Kuisioner Responden Ketiga	145

DAFTAR TABEL

Tabel 3.1 Kebutuhan Fungsional.....	15
Tabel 3.2 Kualitas Perangkat Lunak	18
Tabel 3.3 Kasus Penggunaan.....	19
Tabel 3.4 Spesifikasi Kasus Penggunaan UC-0001	20
Tabel 3.5 Spesifikasi Kasus Penggunaan UC-0002	22
Tabel 3.6 Spesifikasi Kasus Penggunaan UC-0003	24
Tabel 3.7 Spesifikasi Kasus Penggunaan UC-0004	28
Tabel 3.8 Spesifikasi Kasus Penggunaan UC-0005	30
Tabel 3.9 Lingkungan Perancangan Perangkat Lunak	33
Tabel 3.10 Atribut Antarmuka Register Pengguna	37
Tabel 3.11 Atribut Antarmuka Login Pengguna	39
Tabel 3.12 Atribut Antarmuka Beranda	40
Tabel 3.13 Atribut Antarmuka Kompetisi	41
Tabel 3.14 Atribut Antarmuka Detail Peserta	42
Tabel 3.15 Atribut Antarmuka Tentang Bebras	43
Tabel 3.16 Atribut Antarmuka Penjelasan Soal	43
Tabel 3.17 Atribut Antarmuka Soal Pertama	44
Tabel 3.18 Atribut Antarmuka Soal Kedua	46
Tabel 3.19 Atribut Antarmuka Halaman Selesai	48
Tabel 3.20 Atribut Antarmuka Halaman Ranking	48
Tabel 4.1 Lingkungan Implementasi Perangkat Lunak.....	49
Tabel 5.1 Lingkungan Pengujian Fungsionalitas Perangkat Lunak	93
Tabel 5.2 Skenario Uji Coba Fungsionalitas	94
Tabel 5.3 Uji Coba Melihat Tentang Bebras.....	94
Tabel 5.4 Uji Coba Melihat Detail Peserta.....	96
Tabel 5.5 Uji Coba Mengerjakan Soal	98
Tabel 5.6 Uji Coba Melihat Nilai	100
Tabel 5.7 Uji Coba Melihat Ranking	102
Tabel 5.8 Rekapitulasi Hasil Uji Fungsionalitas	104
Tabel 5.9 Hasil Kuesioner Pengguna	104
Tabel 5.10 Rata-Rata Nilai Hasil Kuesioner Pengguna	105

DAFTAR KODE SUMBER

Kode Sumber 4.1 Fungsi Register Pengguna (Bagian Pertama) .	50
Kode Sumber 4.2 Fungsi Register Pengguna (Bagian Kedua)....	51
Kode Sumber 4.3 Fungsi Register Pengguna (Bagian Ketiga) ...	52
Kode Sumber 4.4 Fungsi Register Pengguna (Bagian Keempat)	53
Kode Sumber 4.5 Kontrol Register Pengguna (Bagian Pertama)	53
Kode Sumber 4.6 Kontrol Register Pengguna (Bagian Kedua) ..	54
Kode Sumber 4.7 Kontrol Register Pengguna (Bagian Ketiga) ..	55
Kode Sumber 4.8 Fungsi Login Pengguna (Bagian Pertama).....	56
Kode Sumber 4.9 Fungsi Login Pengguna (Bagian Kedua)	57
Kode Sumber 4.10 Fungsi Login Pengguna (Bagian Ketiga)	58
Kode Sumber 4.11 Kontrol Login Pengguna (Bagian Pertama) .	58
Kode Sumber 4.12 Kontrol Login Pengguna (Bagian Kedua)....	59
Kode Sumber 4.13 Fungsi Detail Peserta (Bagian Pertama).....	61
Kode Sumber 4.14 Fungsi Detail Peserta (Bagian Kedua)	62
Kode Sumber 4.15 Kontrol Detail Peserta (Bagian Pertama)	62
Kode Sumber 4.16 Kontrol Detail Peserta (Bagian Kedua).....	63
Kode Sumber 4.17 Kontrol Jumlah Peserta (Bagian Pertama) ...	63
Kode Sumber 4.18 Kontrol Jumlah Peserta (Bagian Kedua)	64
Kode Sumber 4.19 Fungsi Selesai (Bagian Pertama).....	65
Kode Sumber 4.20 Fungsi Selesai (Bagian Kedua)	66
Kode Sumber 4.21 Fungsi Selesai (Bagian Ketiga)	67
Kode Sumber 4.22 Kontrol Fungsi Selesai (Bagian Pertama)	67
Kode Sumber 4.23 Kontrol Fungsi Selesai (Bagian Kedua)	68
Kode Sumber 4.24 Fungsi Ranking	69
Kode Sumber 4.25 Kontrol Fungsi Ranking	70
Kode Sumber 4.26 Inisiasi Variabel Stack.....	74
Kode Sumber 4.27 Inisiasi Variabel Queue	75
Kode Sumber 4.28 Pop dari Stack.....	75
Kode Sumber 4.29 Push menuju Stack	75
Kode Sumber 4.30 Dequeue dari queue	76
Kode Sumber 4.31 Reset Posisi Queue	76

Kode Sumber 4.32 Enqueue Menuju Queue (Bagian Pertama)....76

Kode Sumber 4.33 Enqueue Menuju Queue (Bagian Kedua)77

Kode Sumber 4.34 Kode Sumber Cek Jawaban (Bagian Pertama)
.....77

Kode Sumber 4.35 Kode Sumber Cek Jawaban (Bagian Kedua)78

Kode Sumber 4.36 Inisiasi Variabel Status Lampu.....78

Kode Sumber 4.37 Inisiasi Variabel Tombol dan Lampu79

Kode Sumber 4.38 Fungsi Switch Pertama79

Kode Sumber 4.39 Fungsi Switch Kedua.....79

Kode Sumber 4.40 Fungsi Switch Ketiga80

Kode Sumber 4.41 Fungsi Switch Keempat.....80

Kode Sumber 4.42 Fungsi Switch Kelima (Bagian Pertama)80

Kode Sumber 4.43 Fungsi Switch Kelima (Bagian Kedua)81

Kode Sumber 4.44 Implementasi Random Soal (Bagian Pertama)
.....81

Kode Sumber 4.45 Implementasi Random Soal (Bagian Kedua)82

Kode Sumber 4.46 Implementasi Random Soal (Bagian Ketiga)83

Kode Sumber 4.47 Implementasi Insert Nilai83

Kode Sumber 4.48 Kontrol Insert Nilai (Bagian Pertama).....84

Kode Sumber 4.49 Kontrol Insert Nilai (Bagian Kedua)85

Kode Sumber 4.50 Implementasi Waktu Hitung Mundur (Bagian
Pertama).....85

Kode Sumber 4.51 Implementasi Waktu Hitung Mundur (Bagian
Kedua)86

Kode Sumber 4.52 Implementasi Waktu Hitung Mundur (Bagian
Ketiga)87

Kode Sumber 4.53 Implementasi Terjemahkan Soal (Bagian
Pertama).....88

Kode Sumber 4.54 Implementasi Terjemahkan Soal (Bagian
Kedua)89

Kode Sumber 4.55 Implementasi Terjemahkan Soal (Bagian
Ketiga).....90

Kode Sumber 4.56 Implementasi Terjemahkan Soal (Bagian
Keempat)91

BAB I

PENDAHULUAN

1.1 Latar Belakang

Kompetisi Bebras didirikan di negara Lithuania oleh Prof. Valentina Dagiene dari University of Vilnius pada tahun 2004. Bebras adalah istilah dalam bahasa Lithuania untuk “beaver” (dalam bahasa Indonesia adalah “berang-berang”). Bebras dipilih sebagai simbol tantangan (*challenge*), karena hewan *beaver* berusaha keras untuk mencapai target secara sempurna dalam aktivitasnya sehari-hari. Mereka membuat bendungan dari ranting-ranting pohon di sungai atau aliran air dan membuat rumahnya sendiri. Kompetisi ini disebut Bebras untuk menunjukkan kerja keras dan kecerdasan diperlukan di dalam kehidupan.

Kompetisi Bebras dilaksanakan setiap tahun. Negara yang sudah berpartisipasi mengikuti Bebras ada 50 negara, belum termasuk Indonesia. Pada tahun 2015, jumlah peserta yang mengikuti Bebras mencapai 1,3 juta siswa dari berbagai belahan dunia. Setelah kunjungannya ke Indonesia pada bulan Februari 2016, Prof. Valentina Dagiene (Vilnius University, Lithuania) yaitu penggagas (*founder*) Bebras Internasional, mengundang Indonesia menjadi *observer* pada *Workshop* Internasional Bebras pada bulan Mei tahun 2016 [1].

Pada Kompetisi Bebras di Indonesia, peserta diminta untuk menjawab soal-soal terkait dengan pemecahan masalah, berpikir komputasional dan kreativitas. Terdapat tiga kategori pada Kompetisi Bebras di Indonesia, yaitu Siaga, Penggalang dan Penegak. Tingkat kesulitan soal disesuaikan dengan kategori dari masing-masing peserta.

Saat ini, Kompetisi Bebras Indonesia mempunyai website yaitu “Olympia.id”, dimana website tersebut menjadi tempat untuk peserta menjawab soal-soal Kompetisi Bebras Indonesia. Namun, pada website tersebut masih mempunyai masalah terutama pada saat menampilkan soal-soal interaktif pada perangkat bergerak.

Dengan aplikasi ini nantinya, diharapkan dapat mengatasi masalah pada website Kompetisi Bebras Indonesia, dimana pengguna dapat melihat dan menjawab soal-soal interaktif pada perangkat bergerak dengan lancar.

1.2 Rumusan Masalah

Rumusan masalah yang diangkat dalam tugas akhir ini dapat dipaparkan sebagai berikut:

1. Bagaimana membuat soal interaktif pada materi kombinatorial?
2. Bagaimana agar soal-soal interaktif tersebut dapat dibuka dengan baik dan tanpa masalah pada perangkat bergerak?

1.3 Batasan Masalah

Permasalahan yang dibahas dalam tugas akhir ini memiliki beberapa batasan antara lain:

1. Metode soal-soal yang akan dibuat adalah metode soal interaktif.
2. Pembuat soal mengerti cara menggunakan aplikasi Unity.

1.4 Tujuan

Tujuan dari pembuatan tugas akhir ini adalah untuk membangun aplikasi media edukasi untuk berpikir secara komputasional pada materi *combinatorial*.

1.5 Manfaat

Manfaat dari hasil pembuatan tugas akhir ini adalah :

1. Soal-soal yang diberikan dapat membuat peserta mengembangkan cara berpikir secara komputasional.
2. Bentuk soal yang menarik dapat memotivasi peserta untuk menyelesaikannya.

3. Menumbuhkan kebiasaan berpikir secara komputasional sejak dini.

1.6 Metodologi

Pembuatan tugas akhir dilakukan menggunakan metodologi sebagai berikut:

A. Studi literatur

Tahap studi literatur merupakan tahap pembelajaran dan pengumpulan informasi yang digunakan untuk mengimplementasikan tugas akhir. Tahap ini diawali dengan pengumpulan literatur, diskusi, eksplorasi teknologi dan pustaka, serta pemahaman dasar teori yang digunakan pada topik tugas akhir. Literatur-literatur yang dimaksud disebutkan sebagai berikut:

1. Kompetisi bebras.
2. WebGL.
3. *Combinatorics*.
4. Unity.
5. Pemrograman C#.
6. Visual Studio.
7. Struktur Data *Stack* dan *Queue*.

B. Perancangan perangkat lunak

Pada tahap ini diawali dengan melakukan analisis awal terhadap permasalahan utama yang muncul pada topik tugas akhir. Kemudian dilakukan perancangan perangkat lunak yang meliputi penentuan data yang akan digunakan dan proses-proses yang akan dilaksanakan. Langkah yang akan digunakan pada tahap ini adalah sebagai berikut:

1. Perancangan soal untuk dikerjakan oleh pengguna.
2. Perancangan sistem untuk mendukung pengerjaan soal.

C. Implementasi dan pembuatan sistem

Pada tahap ini dilakukan implementasi manipulasi data menggunakan database MySQL. *Asset-asset* seperti gambar

untuk pembuatan game didapat dari internet. Selanjutnya aplikasi ini dibangun dengan aplikasi Unity.

D. Uji coba dan evaluasi

Pada tahap ini dilakukan uji coba dengan menggunakan beberapa macam kondisi untuk mencoba aplikasi bisa berjalan atau tidak. Uji fungsionalitas untuk mengetahui apakah aplikasi sudah memenuhi semua kebutuhan fungsional selain itu juga dilakukan pengujian non-fungsionalitas apakah aplikasi sudah bisa memberi soal-soal yang menantang dan kesan *immersive* kepada pengguna. Pengujian aplikasi ini dilakukan dengan metode kuesioner kepada pengguna yang sebelumnya telah diminta untuk menjalankan aplikasi secara langsung.

E. Penyusunan laporan tugas akhir

Pada tahap ini dilakukan penyusunan laporan yang berisi dasar teori, dokumentasi dari perangkat lunak, dan hasil-hasil yang diperoleh selama pengerjaan tugas akhir.

1.7 Sistematika Penulisan

Buku tugas akhir ini terdiri dari beberapa bab yang dijelaskan sebagai berikut:

BAB I PENDAHULUAN

Bab ini berisi latar belakang masalah, rumusan dan batasan permasalahan, tujuan dan manfaat pembuatan tugas akhir, metodologi yang digunakan, dan sistematika penyusunan tugas akhir.

BAB II TINJAUAN PUSTAKA

Bab ini membahas dasar pembuatan dan beberapa teori penunjang yang berhubungan dengan pokok pembahasan yang mendasari pembuatan tugas akhir ini.

BAB III ANALISIS DAN PERANCANGAN

Bab ini membahas analisis dari sistem yang dibuat meliputi analisis permasalahan, deskripsi umum perangkat lunak, spesifikasi kebutuhan, dan identifikasi pengguna. Kemudian membahas rancangan dari sistem yang dibuat meliputi rancangan skenario kasus penggunaan, data, dan antarmuka.

BAB IV IMPLEMENTASI

Bab ini membahas implementasi dari rancangan sistem yang dilakukan pada tahap perancangan. Penjelasan implementasi meliputi implementasi pembuatan soal, implementasi pembuatan aplikasi.

BAB V PENGUJIAN DAN EVALUASI

Bab ini membahas pengujian dari aplikasi yang dibuat dengan melihat keluaran yang dihasilkan oleh aplikasi dan evaluasi untuk mengetahui kemampuan aplikasi.

BAB VI PENUTUP

Bab ini berisi kesimpulan dari hasil pengujian yang dilakukan serta saran untuk pengembangan aplikasi selanjutnya.

BAB II

TINJAUAN PUSTAKA

2.1 Soal Bebras

Bebras adalah inisiatif internasional yang bertujuan untuk mempromosikan informatika (ilmu komputer, atau komputasi) dan berpikir komputasional di kalangan siswa sekolah di segala usia. Peserta biasanya diawasi oleh guru yang dapat mengintegrasikan tantangan Bebras dalam kegiatan mengajar mereka [2].

Yang dilombakan dalam kompetisi adalah sekumpulan soal yang disebut *Bebras task*. *Bebras task* disajikan dalam bentuk uraian persoalan yang dilengkapi dengan gambar yang menarik, sehingga siswa dapat lebih mudah memahami soal. Soal-soal tersebut dapat dijawab tanpa perlu belajar informatika terlebih dahulu, tapi soal tersebut sebetulnya terkait pada konsep tertentu dalam informatika dan *computational thinking* [3].

Penyusun soal Bebras berusaha memilih soal yang menarik untuk memotivasi siswa dalam mengidentifikasi persoalan informatika dan berpikir lebih dalam tentang teknologi. Mereka juga ingin menyajikan sebanyak mungkin topik informatika dan pembelajaran komputer.

2.2 WebGL

WebGL adalah teknologi web yang menyuguhkan akselerasi grafis 3D ke dalam browser tanpa memasang perangkat lunak tambahan. WebGL merupakan kependekan dari Web Graphics Library. Untuk API, WebGL biasanya dipanggil melalui API Javascript dan penggunaannya selalu melibatkan elemen HTML5 <canvas>.

WebGL biasanya digunakan untuk desain web dan game 3D yang berbasis web. WebGL juga juga digunakan oleh

beberapa peneliti untuk tujuan sains. Sebagai contoh, dalam buku yang berjudul "Cellular Automata", peneliti menggunakan WebGL untuk mensimulasikan alur Debris. Sedangkan pada paper "Visualization of molecular structures using state-of-the-art techniques in WebGL", peneliti mensimulasikan molekul didalamnya. NASA juga membangun aplikasi web yang interaktif yang disebut Experience Curiosity untuk merayakan ulang tahun ketiga dimana Curiosity Rover mendarat di Mars [4].

2.3 *Combinatorics*

Combinatorics adalah cabang matematika tentang studi struktur diskrit yang terbatas dan dapat dihitung. Aspek kombinatorik meliputi penghitungan struktur jenis dan ukuran tertentu (*enumerative combinatorics*), menentukan kapan kriteria tertentu dapat dipenuhi, dan membangun dan menganalisa objek yang memenuhi kriteria (seperti pada desain kombinatorial dan teori matroid), menemukan "terbesar", "terkecil", atau "optimal" (kombinasi kombinatorik dan optimalisasi kombinatorial), dan mempelajari struktur kombinatorial yang timbul dalam konteks aljabar, atau menerapkan teknik aljabar untuk masalah kombinatorial [5].

2.4 *Unity*

Unity adalah sebuah *tool* pengembangan *game* yang terintegrasi kuat dengan satu set lengkap alat intuitif dan alur kerja yang cepat untuk membuat 3D interaktif dan konten 2D. Unity merupakan *easy multiplatform publishing*. Unity memiliki *Asset Store* yang menyediakan *Asset* untuk di unduh secara gratis maupun berbayar. Terdapat pula *Unity Community* yang menyediakan tutorial secara gratis untuk semua pengguna unity dalam bahasa inggris. Fitur *scripting* yang disediakan, mendukung 3 bahasa pemrograman, yaitu JavaScript, C#, dan Boo. *Flexible and EasyMoving*, *rotating*, dan *scaling objects* hanya perlu sebaris kode. Begitu juga dengan *Duplicating*,

removing, dan *changing properties*. *Visual Properties Variables* yang di definisikan dengan scripts ditampilkan pada *editor*. Bisa digeser, di *drag and drop*, bisa memilih warna dengan *color picker*. Berbasis .NET. Artinya penjalanan program dilakukan dengan *Open Source .NET platform*, Mono. Serta mendukung pengembangan aplikasi Microsoft, SONY, Qualcomm, BlackBerry, Samsung, Ninetendo, Oculus VR dan Intel [6].

2.5 Pemrograman C#

C# (dibaca: C sharp) merupakan sebuah bahasa pemrograman yang berorientasi objek yang dikembangkan oleh Microsoft sebagai bagian dari inisiatif kerangka .NET Framework. Bahasa pemrograman ini dibuat berbasiskan bahasa C++ yang telah dipengaruhi oleh aspek-aspek ataupun fitur bahasa yang terdapat pada bahasa-bahasa pemrograman lainnya seperti Java, Delphi, Visual Basic, dan lain-lain dengan beberapa penyederhanaan. Menurut standar ECMA-334 C# Language Specification, nama C# terdiri atas sebuah huruf Latin C (U+0043) yang diikuti oleh tanda pagar yang menandakan angka # (U+0023). Tanda pagar # yang digunakan memang bukan tanda kres dalam seni musik (U+266F), dan tanda pagar # (U+0023) tersebut digunakan karena karakter kres dalam seni musik tidak terdapat di dalam keyboard standar.

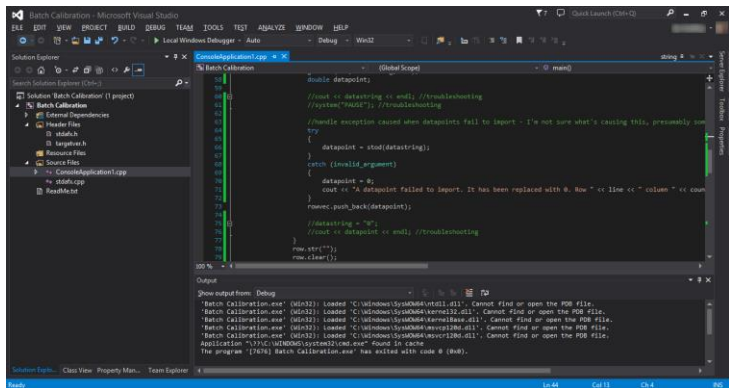
C# didesain untuk memenuhi kebutuhan akan sintaksis C++ yang lebih ringkas dan Rapid Application Development yang 'tanpa batas' (dibandingkan dengan RAD yang 'terbatas' seperti yang terdapat pada Delphi dan Visual Basic).

C# kadang-kadang dapat disebutkan sebagai bahasa pemrograman yang paling mencerminkan dasar dari CLR dimana semua program-program .NET berjalan, dan bahasa ini sangat bergantung pada kerangka tersebut sebab ia secara spesifik didisain untuk mengambil manfaat dari fitur-fitur yang tersedia pada CLR.

Desain dari *Object Oriented* harus berisi objek-objek yang ada pada domain tersebut. Sebagai contoh jika problem domainnya adalah mesin ATM maka objek didalamnya seharusnya adalah *customer*, rekening, dan lain-lain [7].

2.6 Visual Studio

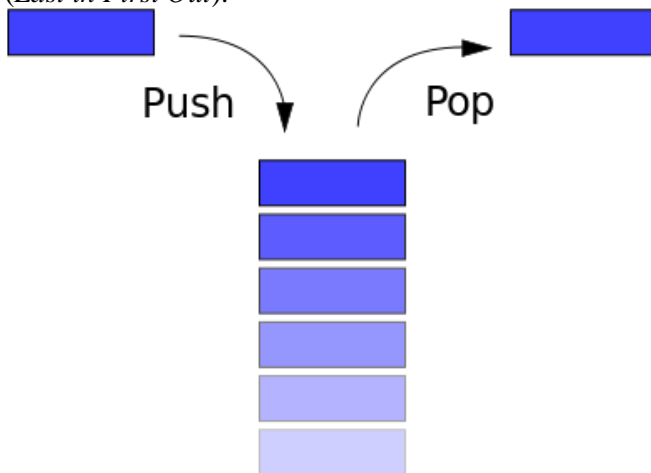
Microsoft Visual Studio merupakan sebuah aplikasi yang terdiri dari kompiler, SDK, *Integrated Development Environment* (IDE), dan dokumentasi yang biasa digunakan untuk mengembangkan aplikasi personal, aplikasi *web*, aplikasi *mobile*, aplikasi konsole maupun komponen aplikasi pada Microsoft Visual Studio. Banyak compiler yang disediakan oleh Microsoft Visual Studio, misalnya saja Visual C++, Visual C#, Visual Basic, Visual Basic .NET, Visual InterDev, Visual J++, Visual J#, Visual FoxPro, dan Visual SourceSafe. Banyak pengembang aplikasi yang mengembangkan aplikasinya menggunakan Microsoft Visual Studio karena banyak fungsi yang telah disediakan oleh Microsoft Visual Studio, sehingga pengembang lebih mudah dan sangat terbantu dalam mengembangkan maupun membuat aplikasi [8]. Tampilan antarmuka aplikasi ini dapat dilihat pada Gambar 2.1.



Gambar 2.1 Tampilan Antarmuka Visual Studio

2.7 Struktur data *stack* dan *queue*

Struktur data *stack* dapat direpresentasikan sebagai sebuah tumpukan. Implementasi dasar pada *stack* dapat disebut dengan LIFO (*Last in First Out*).



Gambar 2.2 Struktur data *stack*

Struktur data *queue* juga direpresentasikan dengan tumpukan namun dengan teori dasar yang berbeda yaitu FIFO (*First in First Out*). Beberapa operasi yang ada pada *queue*, yaitu: *enqueue*, *dequeue*, *get-size*, *is-empty*, *is-full*.

2.8 Aplikasi Terkait yang Sudah Ada

Selama beberapa tahun terakhir, telah dikembangkan aplikasi perangkat bergerak untuk melatih berpikir secara komputasional. Salah satunya bernama *lightbot*. *Lightbot* adalah aplikasi edukasi untuk anak-anak yang mengenalkan beberapa dasar dari pemrograman. Aplikasi ini tersedia pada perangkat iOS dan Android [9]. Untuk tampilan aplikasi *lightbot* dapat dilihat pada gambar 2.3.



Gambar 2.3 Tampilan Antarmuka *lightbot*

BAB III

ANALISIS DAN PERANCANGAN

Bab ini membahas analisis kebutuhan dan rancangan yang akan digunakan untuk membangun perangkat lunak yang diajukan sebagai tugas akhir.

3.1 Analisis Sistem

Tahap analisis dibagi menjadi beberapa bagian, antara lain analisis permasalahan, deskripsi umum sistem, dan spesifikasi kebutuhan perangkat lunak.

3.1.1 Analisis Permasalahan

Bebras merupakan sebuah inisiatif untuk memperkenalkan cara berpikir komputasional khususnya kepada para pelajar dan masyarakat luas. Kompetisi yang terdapat pada bebras menyajikan soal-soal yang menerapkan aspek ilmu komputasi atau informatika yang dimaksudkan agar peserta dapat melatih berpikir secara logis, kreatif dan kritis dengan menerapkan konsep berpikir komputasional untuk menyelesaikan soal-soal yang diberikan.

Setiap peserta dapat mengerjakan soal-soal secara interaktif pada sistem. Soal-soal yang diberikan dilengkapi dengan berbagai gambar menarik agar memotivasi serta memudahkan peserta dalam memahami dan mengerjakan soal. Peserta dapat menjawab soal secara interaktif sesuai dengan pertanyaan yang telah diberikan. Setiap soal memiliki batasan waktu yang berbeda-beda. Pada saat kompetisi bebras berlangsung terdapat alat pengatur waktu yang akan dihitung secara mundur sesuai dengan batasan waktu yang telah ditentukan. Nilai peserta akan menentukan peringkat yang akan di dapatkan.

3.1.2 Deskripsi Umum Sistem

Pada tampilan halaman web terdapat tentang bebras dan detail peserta yang dapat dilihat oleh para peserta. Tentang bebras berisi informasi mengenai bebras dan detail peserta berisi informasi mengenai peserta secara rinci. Para peserta dapat melakukan kompetisi bebras dengan melakukan registrasi dan login terlebih dahulu, kemudian peserta dapat mengerjakan soal-soal di dalam kompetisi bebras secara online.

Soal-soal yang diberikan memiliki urutan nomor secara acak. Peserta dapat menjawab soal secara interaktif sesuai dengan pertanyaan yang telah diberikan. Terdapat dua bahasa yang tersedia ketika mengerjakan soal yaitu bahasa Indonesia dan bahasa Inggris, peserta dapat memilih salah satu bahasa yang telah tersedia tersebut. Setiap soal memiliki batasan waktu yang berbeda-beda. Pada saat kompetisi bebras berlangsung terdapat alat pengatur waktu yang akan dihitung secara mundur sesuai dengan batasan waktu yang telah ditentukan. Peserta yang telah menyelesaikan semua soal dapat melihat nilai serta peringkat yang didapatkan.

3.1.3 Spesifikasi Kebutuhan Perangkat Lunak

Sesuai dengan uraian perangkat lunak yang telah dijelaskan pada bagian deskripsi umum sistem, dibutuhkan adanya spesifikasi perangkat lunak agar dapat memberikan solusi dari permasalahan yang diberikan dan dapat bekerja dengan baik dalam mengakomodasi kebutuhan. Spesifikasi kebutuhan perangkat lunak pada tugas akhir ini terdiri dari kebutuhan fungsional, kebutuhan non-fungsional, aktor, dan kasus penggunaan.

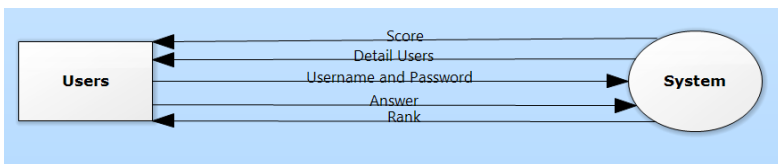
3.1.3.1 Kebutuhan Fungsional

Tabel 3.1 Kebutuhan Fungsional

No	Kebutuhan Fungsional	Deskripsi
1	Melihat tentang bebras	Menampilkan informasi mengenai bebras
2	Melihat detail peserta	Menampilkan data peserta dan total peserta
3	Menjalankan aktifitas mengerjakan soal	Menjalankan setiap aktivitas mengerjakan soal untuk peserta
4	Melihat nilai	Menampilkan nilai yang didapatkan peserta
5	Melihat peringkat	Menampilkan peringkat yang diperoleh peserta

3.1.3.2 Aliran Informasi

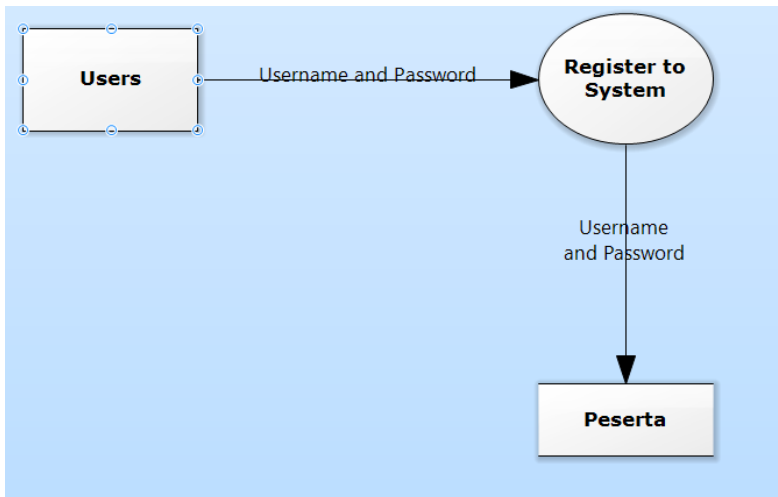
Aliran Informasi untuk sistem dapat dilihat pada Gambar 3.1 berupa *data flow diagram level 0*. Dalam diagram tersebut dapat dilihat bahwa pengguna dapat mengakses aplikasi dengan memasukkan data-data yang diperlukan untuk *login* terlebih dahulu.



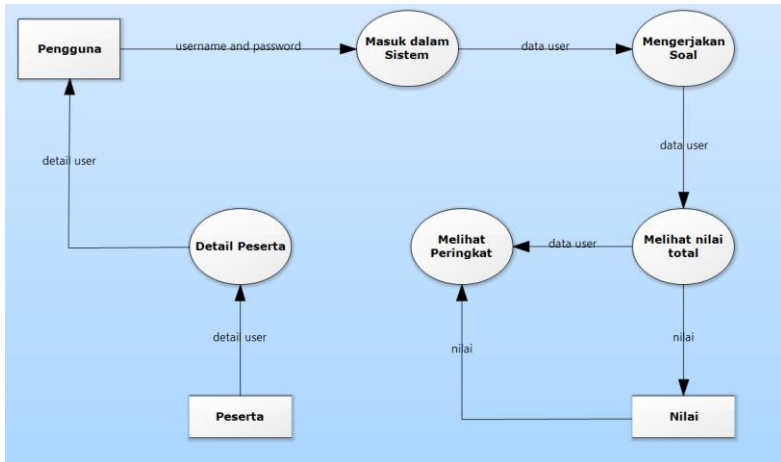
Gambar 3.1 DFD Level 0

3.1.3.3 DFD Level 1

Dalam DFD level 1, secara keseluruhan terbagi menjadi dua bagian besar, yaitu ketika pengguna masuk atau *login* dan ketika pengguna melakukan registrasi. Setelah melakukan proses registrasi, pengguna dapat masuk dan mengakses aplikasi untuk mengerjakan soal. DFD level 1 dapat dilihat pada Gambar 3.2 dan Gambar 3.3.



Gambar 3.2 DFD Level 1 untuk Proses Registrasi



Gambar 3.3 DFD Level 1

3.1.3.4 Kebutuhan Non-Fungsional

Kebutuhan non-fungsional yang harus dipenuhi oleh sistem sebagai berikut:

1. Kebutuhan Performa
Perangkat lunak akan berjalan dengan performa terbaik jika dijalankan diatas spesifikasi minimal.
2. Kebutuhan Perlindungan Keamanan
Username dan *password* dibutuhkan untuk mengamankan data pengguna yang berhak mengakses sistem sebagai syarat memasuki sistem dan melakukan semua fungsionalitas sistem.
3. Kualitas perangkat lunak dapat dilihat pada Tabel 3.2.

Tabel 3.2 Kualitas Perangkat Lunak

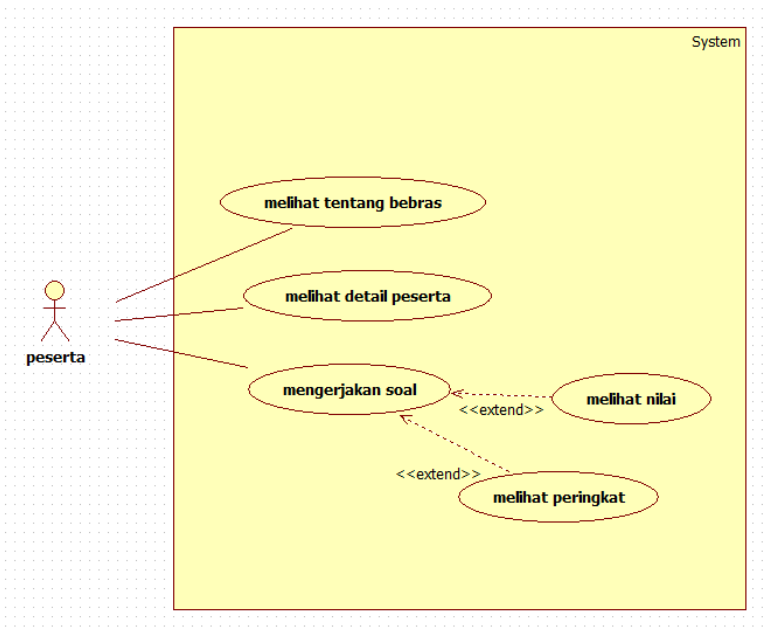
No	Parameter	Deskripsi
1	Usabilitas	Pada aspek ini indikator yang digunakan adalah kemudahan penggunaan menu, kemudahan dalam mendapatkan informasi, dan kesan <i>immersive</i>
2	Tingkat kualitas	Aplikasi dibangun dengan antarmuka pengguna yang konsisten, mudah dipahami dan mudah dioperasikan
3	Operasional	Aspek operasional yang mempengaruhi aplikasi ini berkaitan dengan lingkungan pengoperasional, yaitu meliputi jumlah <i>core</i> CPU perangkat, RAM, kecepatan prosesor dll.
4	Bahasa penyampaian	Terdapat dua bahasa yang tersedia ketika mengerjakan soal yaitu bahasa Indonesia dan bahasa Inggris, peserta dapat memilih salah satu bahasa yang telah tersedia tersebut.

3.1.3.5 Aktor

Pengertian pengguna adalah pihak-pihak yang terlibat dan berinteraksi secara langsung dengan sistem. Pengguna pada aplikasi ini adalah peserta. Peserta merupakan seseorang yang menjalankan aplikasi ini.

3.1.3.6 Kasus Penggunaan

Berdasarkan analisis spesifikasi kebutuhan fungsional dan analisis aktor dari sistem, dibuat kasus penggunaan sistem. Kasus penggunaan digambarkan dalam tabel penjelasan kasus penggunaan dan diagram kasus penggunaan. Tabel penjelasan kasus penggunaan dapat dilihat pada Tabel 3.3.



Gambar 3.4 Diagram Kasus Penggunaan

Tabel 3.3 Kasus Penggunaan

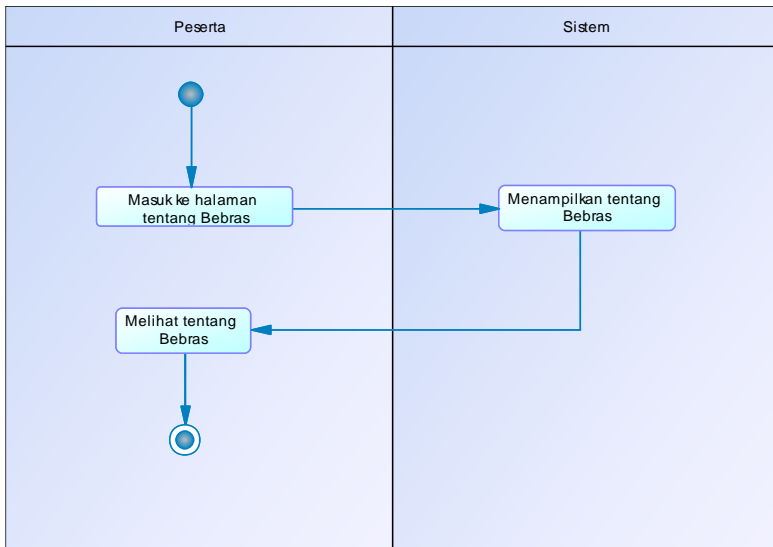
Kode Kasus Penggunaan	Nama	Aktor
UC-0001	Melihat tentang bebras	Peserta
UC-0002	Melihat detail peserta	Peserta
UC-0003	Mengerjakan soal	Peserta
UC-0004	Melihat nilai	Peserta
UC-0005	Melihat peringkat	Peserta

3.1.3.6.1 Melihat Tentang Bebras (UC-0001)

Kasus penggunaan nomor UC-0001 ini diakses ketika peserta hendak melihat informasi tentang bebras pada sistem. Spesifikasi, diagram aktivitas, dan sekuens kasus penggunaan ini dapat dilihat pada Tabel 3.4.

Tabel 3.4 Spesifikasi Kasus Penggunaan UC-0001

Kode Use Case	UC-0001	
Nama Use Case	Melihat tentang bebras	
Aktor	Peserta	
Deskripsi	Peserta dapat melihat tentang bebras	
Relasi	-	
Kondisi Awal	Sistem belum menampilkan informasi tentang bebras	
Kondisi Akhir	Sistem sudah menampilkan informasi tentang bebras	
Alur kejadian normal	Aktor	Sistem
	1. Memilih pilihan “Tentang Bebras”	
		2. Menampilkan halaman yang berisi informasi tentang bebras
Alur kejadian alternatif	Aktor	Sistem



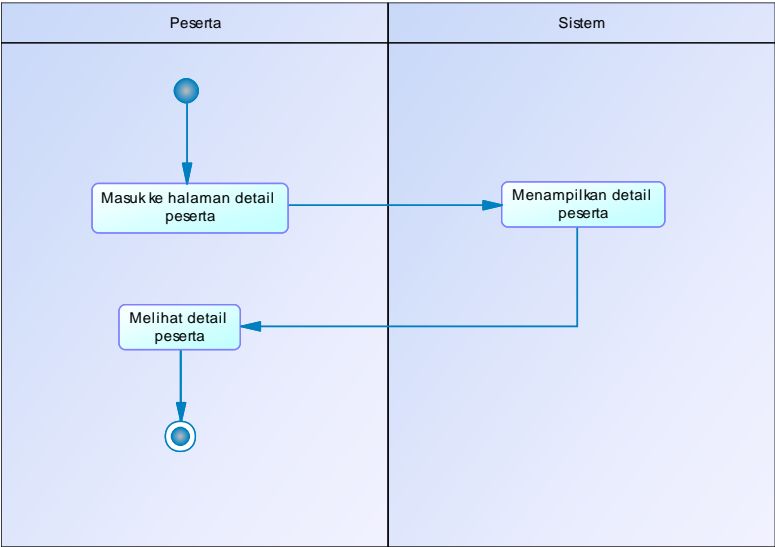
Gambar 3.5 Diagram Aktifitas Melihat Tentang Bebras

3.1.3.6.2 Melihat Detail Peserta (UC-0002)

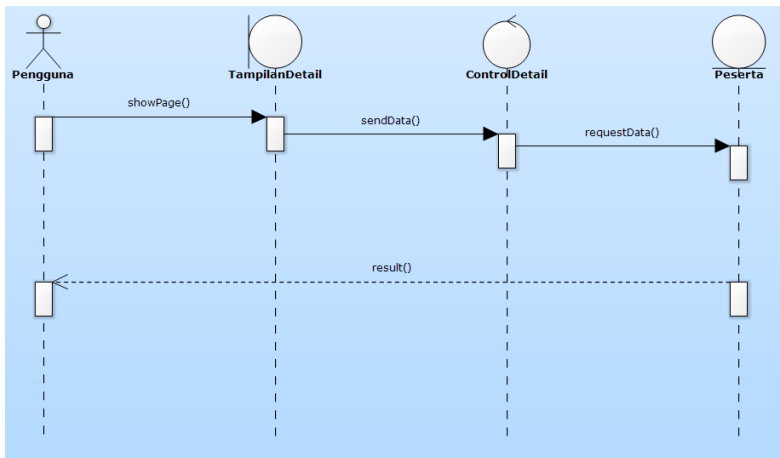
Kasus penggunaan nomor UC-0002 ini diakses ketika peserta hendak melihat detail peserta pada sistem. Spesifikasi, diagram aktivitas, dan sekuens kasus penggunaan ini dapat dilihat pada Tabel 3.5.

Tabel 3.5 Spesifikasi Kasus Penggunaan UC-0002

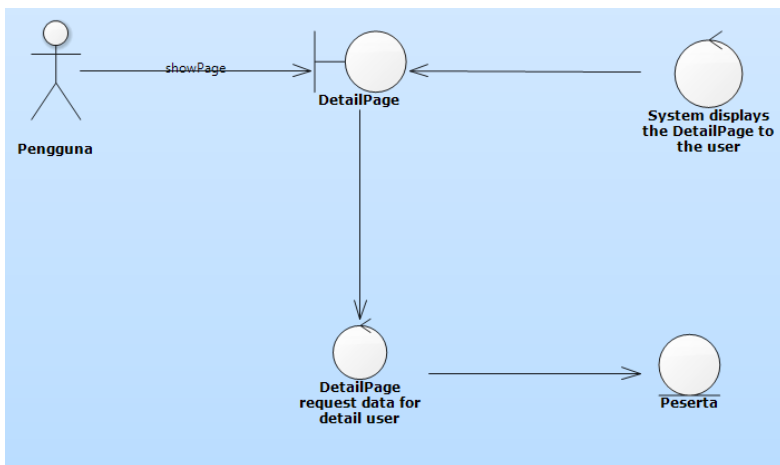
Kode Use Case	UC-0002	
Nama Use Case	Melihat detail peserta	
Aktor	Peserta	
Deskripsi	Peserta dapat melihat detail peserta	
Relasi	-	
Kondisi Awal	Sistem belum menampilkan data detail peserta	
Kondisi Akhir	Sistem sudah menampilkan data detail peserta	
Alur kejadian normal	Aktor	Sistem
	1. Memilih pilihan “Detail Peserta”	
		2. Menampilkan halaman yang berisi data detail peserta
Alur kejadian alternative	Aktor	Sistem



Gambar 3.6 Diagram Aktifitas Melihat Detail Peserta



Gambar 3.7 Diagram Sekuens Melihat Detail Peserta



Gambar 3.8 Robustness Diagram Melihat Detail Peserta

3.1.3.6.3 Mengerjakan Soal (UC-0003)

Kasus penggunaan nomor UC-0003 ini diakses ketika peserta hendak mengerjakan soal yang terdapat pada sistem.

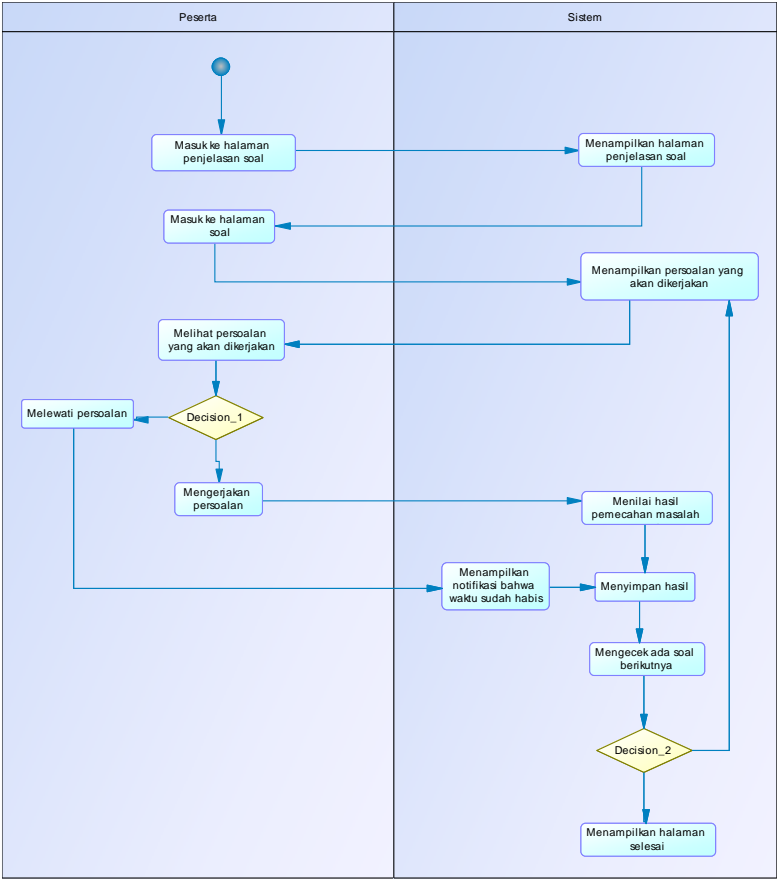
Spesifikasi, diagram aktivitas, dan sekuens kasus penggunaan ini dapat dilihat pada Tabel 3.6.

Tabel 3.6 Spesifikasi Kasus Penggunaan UC-0003

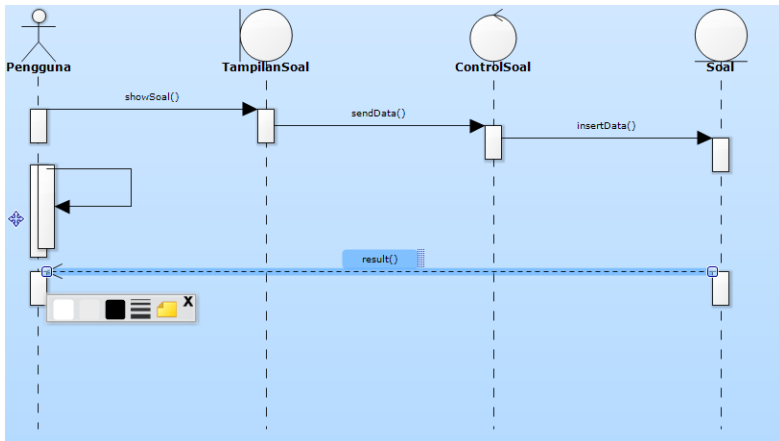
Kode Use Case	UC-0003	
Nama Use Case	Mengerjakan Soal	
Aktor	Peserta	
Deskripsi	Peserta dapat mengerjakan soal yang terdapat di sistem	
Relasi	-	
Kondisi Awal	Peserta belum dapat mengerjakan soal	
Kondisi Akhir	Peserta sudah dapat mengerjakan soal	
Alur kejadian normal	Aktor	Sistem
	1. Memilih pilihan “Kompetisi”	
		2. Menampilkan halaman yang berisi tombol “Login”
	3. Menekan tombol “Login”	
		4. Menampilkan form Login yang berisi input “Username” dan “Password”
	5.a. Memilih “Login”	
		6. Menampilkan halaman yang berisi penjelasan tentang soal dan tombol “Mulai”
	7. Menekan “Mulai”	
		8. Menampilkan halaman yang berisi soal serta pilihan jawaban dan tombol “selanjutnya”, “bahasa”, “Inggris”

Alur kejadian normal	Aktor	Sistem
	9.a.Memilih “selanjutnya”	
		10.Menyimpan nilai kedalam <i>database</i>
Alur kejadian alternative	Aktor	Sistem
	5.b.Memilih “Register”	
		5.b.1. Menampilkan form Register yang berisi input “Username”, “Nama”, “Password”, “Ulangi Password” dan <i>Dropdown</i> “Pilih Provinsi Anda”
	5.b.2. Memilih “Register”	
		5.b.3 Menyimpan data yang baru ke dalam <i>database</i>
	5.b.4 Memilih “Kembali”	
		5.b.5 Menampilkan form Login yang berisi input “Username” dan “Password”
	5.c.Memilih “Home”	
		5.c.1.Menampilkan halaman beranda
	9.b.Memilih “bahasa”	
		9.b.1 Menampilkan soal dalam bahasa Indonesia

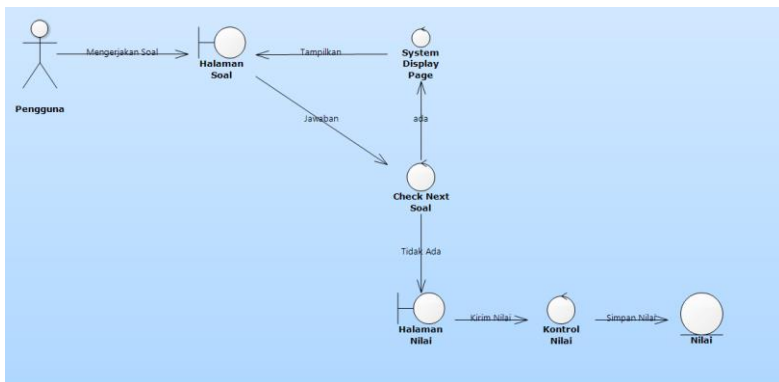
Alur kejadian alternative	Aktor	Sistem
	9.b.2.Memilih Inggris	
		9.b.3 Menampilkan soal dalam bahasa Inggris



Gambar 3.9 Diagram Aktivitas Mengerjakan Soal



Gambar 3.10 Diagram Sekuens Mengerjakan Soal



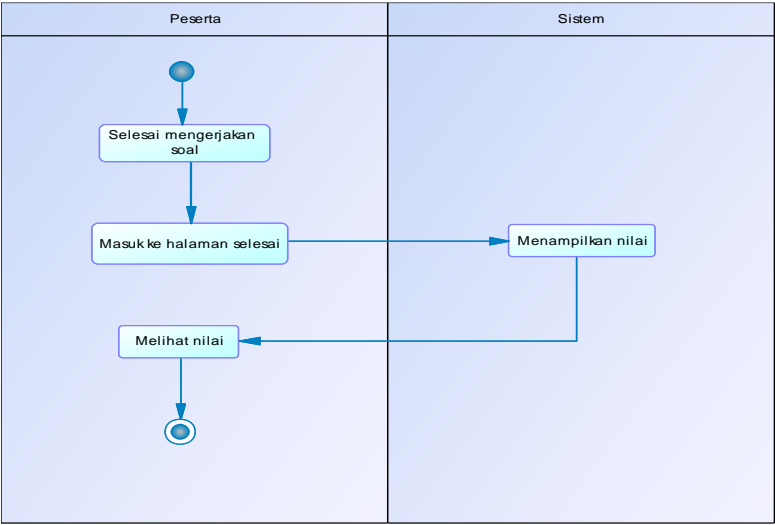
Gambar 3.11 Robustness Diagram Mengerjakan Soal

3.1.3.6.4 Melihat Nilai (UC-0004)

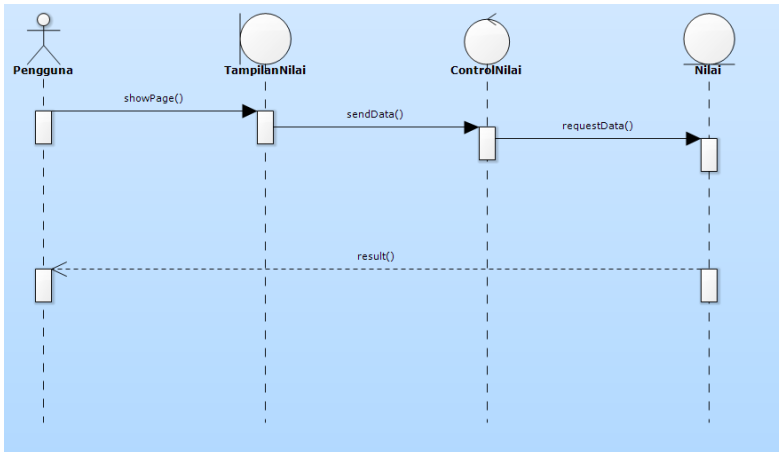
Kasus penggunaan nomor UC-0004 ini diakses ketika peserta ingin melihat nilai pada sistem. Spesifikasi, diagram aktivitas, dan sekuens kasus penggunaan ini dapat dilihat pada Tabel 3.7.

Tabel 3.7 Spesifikasi Kasus Penggunaan UC-0004

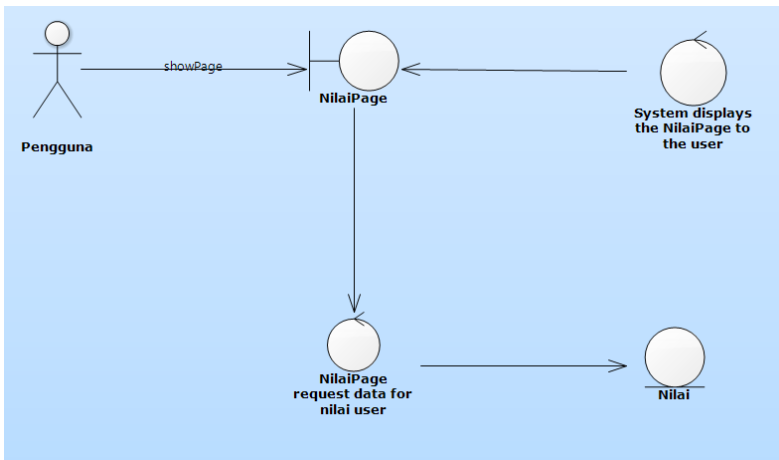
Kode Use Case	UC-0004	
Nama Use Case	Melihat nilai	
Aktor	Peserta	
Deskripsi	Peserta dapat melihat nilai	
Relasi	-	
Kondisi Awal	Sistem belum dapat menampilkan data nilai	
Kondisi Akhir	Sistem sudah dapat menampilkan data nilai	
Alur kejadian normal	Aktor	Sistem
	1. Memilih pilihan “Selanjutnya”	
		2. Menampilkan halaman yang berisi nilai
Alur kejadian alternative	Aktor	Sistem



Gambar 3.12 Diagram Aktivitas Melihat Nilai



Gambar 3.13 Diagram Sekuens Melihat Nilai



Gambar 3.14 Robustness Diagram Melihat Nilai

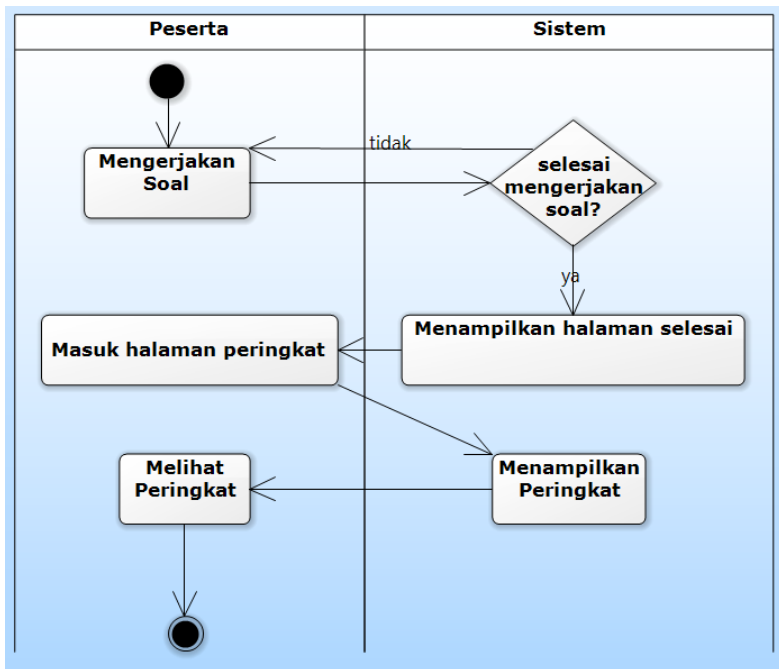
3.1.3.6.5 Melihat Ranking (UC-0005)

Kasus penggunaan nomor UC-0005 ini diakses ketika peserta hendak melihat ranking pada sistem. Spesifikasi, diagram

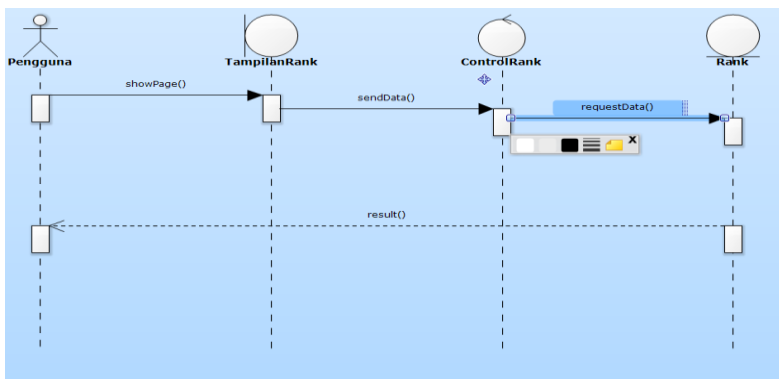
aktivitas, dan sekuens kasus penggunaan ini dapat dilihat pada Tabel 3.8.

Tabel 3.8 Spesifikasi Kasus Penggunaan UC-0005

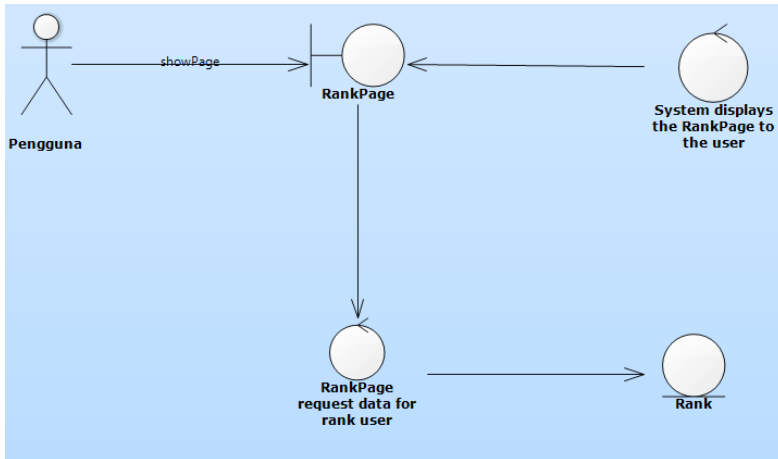
Kode Use Case	UC-0005	
Nama Use Case	Melihat ranking	
Aktor	Peserta	
Deskripsi	Peserta dapat melihat ranking	
Relasi	-	
Kondisi Awal	Sistem belum menampilkan data ranking	
Kondisi Akhir	Sistem sudah menampilkan data ranking	
Alur kejadian normal	Aktor	Sistem
	1.a. Memilih pilihan “Ranking”	
Alur kejadian alternative		2. Menampilkan halaman yang berisi data peringkat
	Aktor	Sistem
	1.b. Memilih pilihan “Log Out”	
		1.c. Menampilkan halaman beranda



Gambar 3.15 Diagram Aktifitas Melihat *Ranking*



Gambar 3.16 Diagram Sekuens Melihat *Ranking*



Gambar 3.17 Robustness Diagram Melihat Ranking

3.2 Perancangan

Perancangan dalam subbab ini membahas perancangan dari aplikasi tugas akhir. Subbab ini terdiri dari lingkungan perancangan perangkat lunak, perancangan arsitektur sistem, perancangan diagram kelas, perancangan basis data, dan perancangan antarmuka pengguna.

3.2.1 Lingkungan Perancangan Perangkat Lunak

Spesifikasi perangkat keras serta perangkat lunak yang digunakan dalam tahap perancangan perangkat lunak tugas akhir ini seperti dijelaskan pada Tabel 3.9.

Tabel 3.9 Lingkungan Perancangan Perangkat Lunak

Perangkat Keras	Komputer	HP Pavilion Sleekbook 15
	Prosesor	Intel® Core™ i5-3230M CPU @ 2.60GHz (2.60GHz)
	Memori Primer	12GB
	Memori Sekunder	1TB
Perangkat Lunak	Sistem Operasi	Windows 10 Home 64-bit
	Perangkat Lunak	Unity 5.5.2, Paint, Sybase PowerDesigner 16.5, Microsoft Word 2013

3.2.2 Perancangan Arsitektur Sistem

**Gambar 3.18 Arsitektur Sistem**

Pada arsitektur sistem ini, untuk mengakses aplikasi ini membutuhkan *browser* seperti mozilla dan chrome dan terhubung ke internet.

Soal yang dibuat menggunakan *game engine* unity kemudian di *build* ke dalam HTML5 agar bisa dibuka dalam bentuk web. Kemudian melalui web service berupa PHP dilakukan penyimpanan ke dalam database. Dalam proses aktivitas aplikasi, aplikasi dapat dibuka dalam bentuk web, kemudian data akan disimpan melalui web service berupa PHP ke dalam database. Perancangan arsitektur sistem pada aplikasi ini dapat dilihat pada Gambar 3.7.

3.2.3 Perancangan Struktur Data

Dalam membuat suatu aplikasi perangkat bergerak, diperlukan analisis kebutuhan berupa perancangan basis data. Basis data yang digunakan adalah MySQL. Rancangan basis data ditampilkan dalam bentuk *Conceptual Data Model* (CDM) dan *Physical Data Model* (PDM). Penjelasan lebih lengkap berupa CDM dan PDM terdapat pada Gambar 3.19 dan Gambar 3.20.

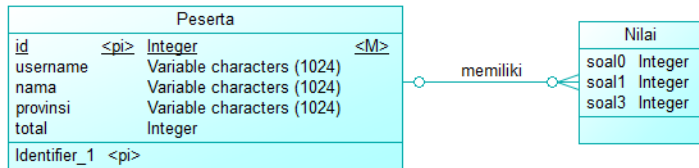
3.2.3.1 Tabel Peserta

Tabel Peserta adalah tabel yang digunakan untuk menyimpan data-data para. Tabel ini mempunyai atribut-atribut, yaitu id, *username*, nama, *password*, nilai, dan provinsi.

3.2.3.2 Tabel Nilai

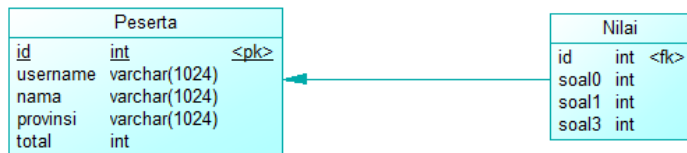
Tabel Nilai adalah tabel yang digunakan untuk menyimpan data-data nilai. Tabel ini merupakan yang mempunyai atribut-atribut, yaitu id, soal0, soal1, soal2.

3.2.3.3 Conceptual Data Modelling (CDM)



Gambar 3.19 Conceptual Data Modelling

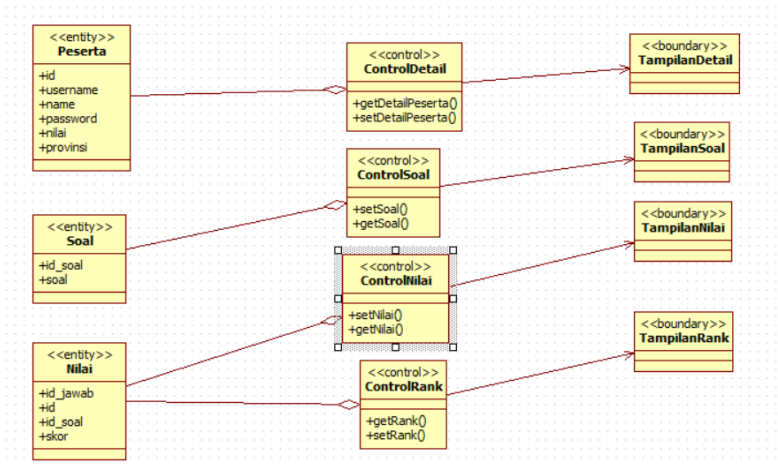
3.2.3.4 Physical Data Modelling (PDM)



Gambar 3.20 Physical Data Modeling

3.2.4 Perancangan Kelas Diagram

Perancangan diagram kelas berisi rancangan dari kelas-kelas yang digunakan untuk membangun sistem. Pada subbab ini, hubungan dan perilaku antar kelas digambarkan dengan lebih jelas. Terdapat entitas peserta, soal dan nilai yang berhubungan dengan masing-masing kontrol yaitu detail, soal, nilai dan *rank*. Selengkapnya dapat dilihat pada Gambar 3.21.



Gambar 3.21 Diagram Kelas Aplikasi

3.2.5 Perancangan Antarmuka Pengguna

Perancangan antarmuka pengguna merupakan hal yang penting dalam melakukan perancangan perangkat lunak. Antarmuka pengguna yang berhubungan langsung dengan aktor harus memiliki kemudahan-kemudahan dan tampilan yang menarik bagi penggunanya. Berdasarkan pengguna yang menggunakan aplikasi, maka terdapat antarmuka untuk peserta.

3.2.5.1 Rancangan Halaman Antarmuka *Register* Pengguna

Halaman ini digunakan oleh pengguna pengguna untuk bisa melakukan pengerjaan soal. Pengguna harus mengisi isian *username*, nama, *password*, ulangi *password*, dan memilih *dropdown* provinsi untuk dapat masuk ke dalam sistem. Atribut antarmuka register dapat dilihat pada Gambar 3.17 dan dijelaskan pada Tabel 3.10

Gambar 3.22 Rancangan Antarmuka *Register*

Tabel 3.10 Atribut Antarmuka *Register* Pengguna

No.	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan
1	<i>Username</i>	<i>InputField</i>	Input untuk memasukkan <i>username</i> pengguna
2	Nama	<i>InputField</i>	Input untuk memasukkan nama pengguna
3	<i>Password</i>	<i>InputField</i>	Input untuk memasukkan <i>password</i>
4	<i>RePassword</i>	<i>InputField</i>	Input untuk memasukkan ulang <i>password</i>
5	Provinsi	<i>Dropdown</i>	Input untuk memilih provinsi
6	<i>btnRegister</i>	<i>Button</i>	Tombol aksi untuk menyimpan data ke database
7	<i>btnKembali</i>	<i>Button</i>	Tombol aksi untuk menuju halaman login

3.2.5.2 Rancangan Halaman Antarmuka *Login* Pengguna

Halaman ini digunakan oleh pengguna untuk bisa melakukan pengerjaan soal. Pengguna harus mengisi isian *username* dan *password* untuk dapat masuk ke dalam sistem. Atribut antarmuka login dapat dilihat pada Tabel 3.11.

LOGIN

Username 1

Password 2

Login 3

Belum punya akun? klik tombol Register

Register 4

Beranda 5

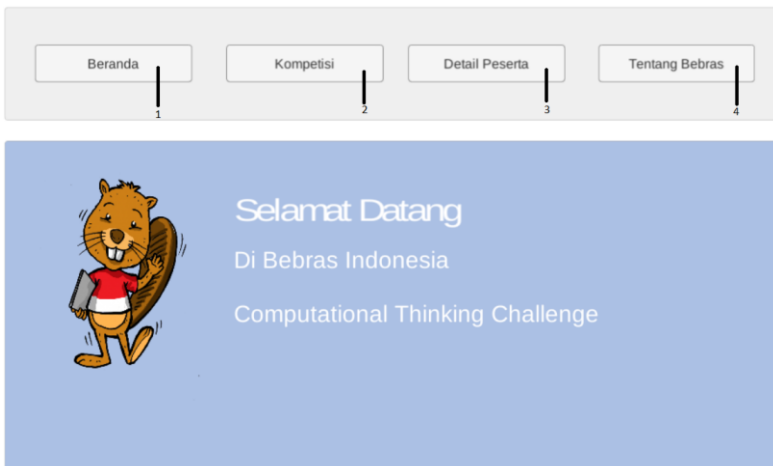
Gambar 3.23 Rancangan Antarmuka *Login*

Tabel 3.11 Atribut Antarmuka Login Pengguna

No.	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan
1	<i>Username</i>	<i>InputField</i>	Input untuk memasukkan <i>username</i> pengguna
2	<i>Password</i>	<i>InputField</i>	Input untuk memasukkan <i>password</i> pengguna
3	<i>btnLogin</i>	<i>Button</i>	Tombol aksi untuk menuju ke halaman Penjelasan soal
4	<i>btnRegis</i>	<i>Button</i>	Tombol aksi untuk menuju ke halaman Register
5	<i>btnKembaliHome</i>	<i>Button</i>	Tombol aksi untuk menuju ke halaman Beranda

3.2.5.3 Rancangan Halaman Antarmuka Beranda

Halaman ini akan ditampilkan pada sistem saat pengguna pertama kali membuka web. Atribut antarmuka beranda dapat dilihat pada Tabel 3.12

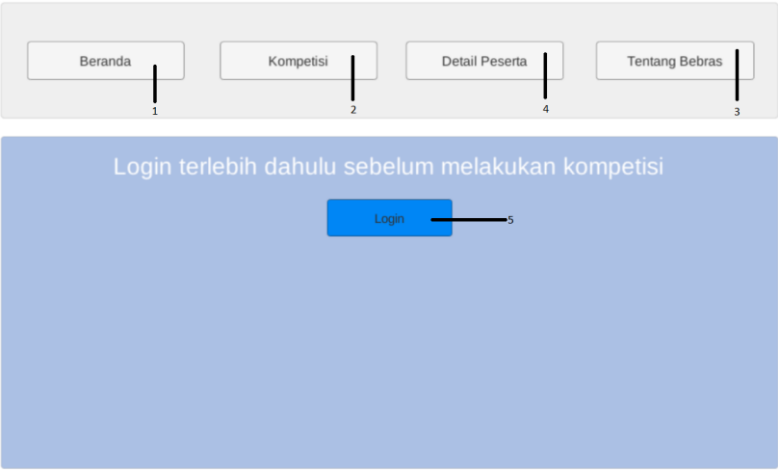
**Gambar 3.24 Rancangan Antarmuka Beranda**

Tabel 3.12 Atribut Antarmuka Beranda

No.	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan
1	<i>btnHome</i>	<i>Button</i>	Tombol aksi untuk menuju ke halaman beranda
2	<i>btnKompetisi</i>	<i>Button</i>	Tombol aksi untuk menuju ke halaman kompetisi
3	<i>btnInfo</i>	<i>Button</i>	Tombol aksi untuk menuju ke halaman tentang bebras
4	<i>btnDetailPeserta</i>	<i>Button</i>	Tombol aksi untuk menuju ke halaman detail peserta

3.2.5.4 Rancangan Halaman Antarmuka Kompetisi

Halaman ini akan ditampilkan untuk masuk kompetisi melalui *login* terlebih dahulu. Atribut antarmuka kompetisi dapat dilihat pada Tabel 3.13.



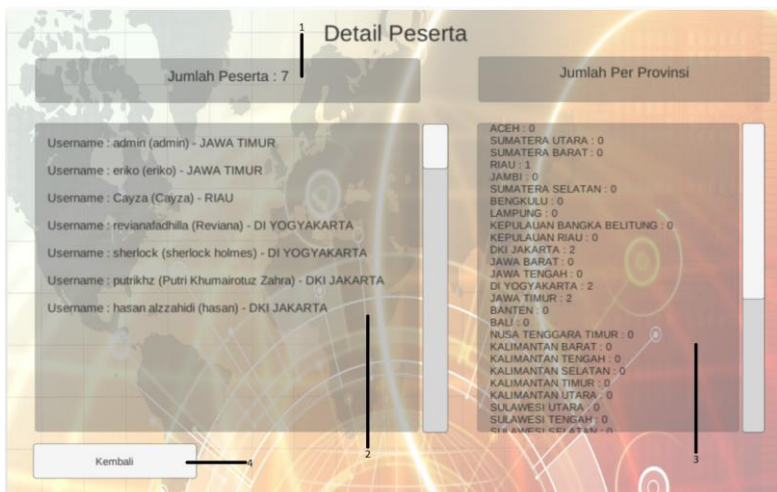
Gambar 3.25 Rancangan Antarmuka Kompetisi

Tabel 3.13 Atribut Antarmuka Kompetisi

No.	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan
1	<i>btnHome</i>	<i>Button</i>	Tombol aksi untuk menuju ke halaman beranda
2	<i>btnKompetisi</i>	<i>Button</i>	Tombol aksi untuk menuju ke halaman kompetisi
3	<i>btnInfo</i>	<i>Button</i>	Tombol aksi untuk menuju ke halaman tentang bebras
4	<i>btnDetailPeserta</i>	<i>Button</i>	Tombol aksi untuk menuju ke halaman detail peserta
5	<i>btnMauLogin</i>	<i>Button</i>	Tombol aksi untuk menuju ke halaman Login

3.2.5.5 Rancangan Halaman Detail Peserta

Pada halaman ini akan ditampilkan data peserta, jumlah peserta dan jumlah per provinsi. Atribut antarmuka detail peserta dapat dilihat pada Tabel 3.14.



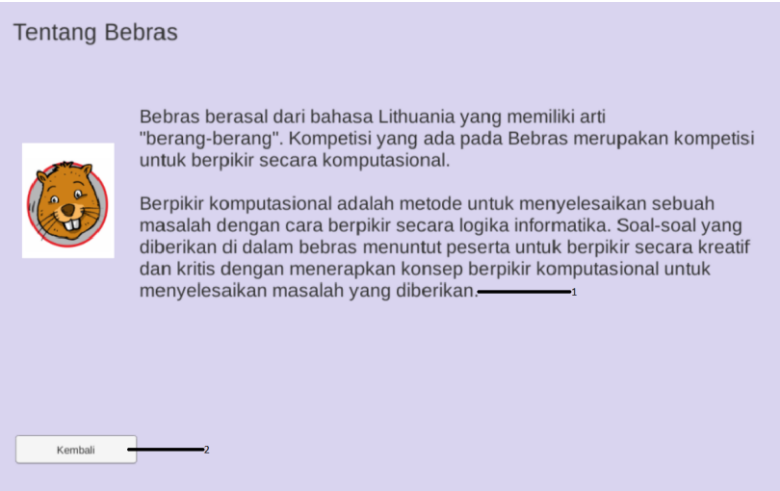
Gambar 3.26 Rancangan Antarmuka Detail Peserta

Tabel 3.14 Atribut Antarmuka Detail Peserta

No.	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan
1	<i>jmlpeserta</i>	<i>Text</i>	Informasi jumlah peserta
2	<i>Usernamepeserta</i>	<i>Text</i>	Informasi <i>username</i> , nama, provinsi dari peserta
3	<i>prov</i>	<i>Text</i>	Informasi jumlah per provinsi
4	<i>btnKembali</i>	<i>Button</i>	Tombol aksi untuk menuju ke halaman Beranda

3.2.5.6 Rancangan Halaman Antarmuka Tentang Bebras

Pada halaman ini akan ditampilkan informasi tentang bebras. Atribut antarmuka tentang bebras dapat dilihat pada Tabel 3.15.



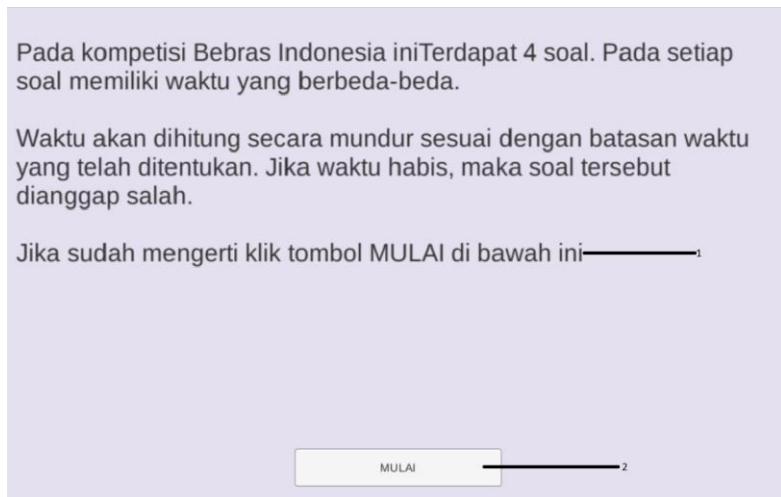
Gambar 3.27 Rancangan Antarmuka Tentang Bebras

Tabel 3.15 Atribut Antarmuka Tentang Bebras

No.	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan
1	<i>Text</i>	<i>Text</i>	Informasi mengenai bebras
2	<i>btnKembali</i>	<i>Button</i>	Tombol aksi untuk menuju halaman Beranda

3.2.5.7 Rancangan Halaman Antarmuka Penjelasan Soal

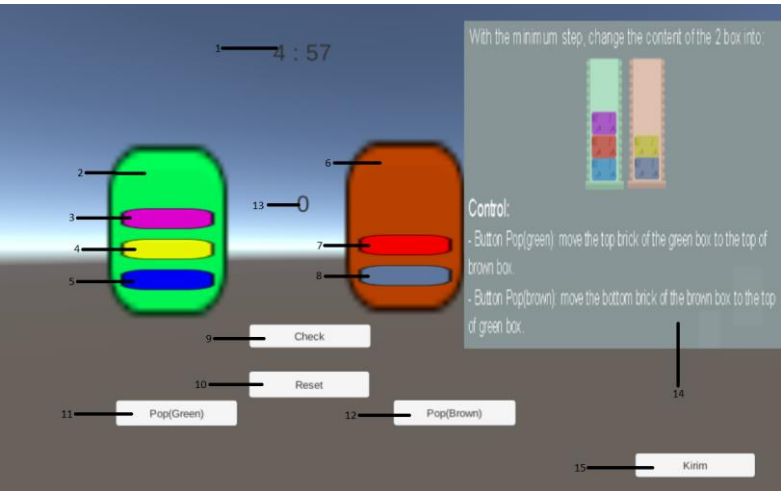
Pada halaman ini akan ditampilkan informasi tentang penjelasan soal. Atribut antarmuka dapat dilihat pada Tabel 3.16.

**Gambar 3.28 Rancangan Antarmuka Penjelasan Soal****Tabel 3.16 Atribut Antarmuka Penjelasan Soal**

No.	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan
1	<i>Text</i>	<i>Text</i>	Informasi tentang penjelasan soal
2	<i>btnno1</i>	<i>Button</i>	Tombol aksi untuk menuju halaman Soal

3.2.5.8 Rancangan Halaman Antarmuka Soal Pertama

Pada halaman ini akan ditampilkan uraian soal. Atribut antarmuka dapat dilihat pada Tabel 3.17.



Gambar 3.29 Rancangan Antarmuka Soal Pertama

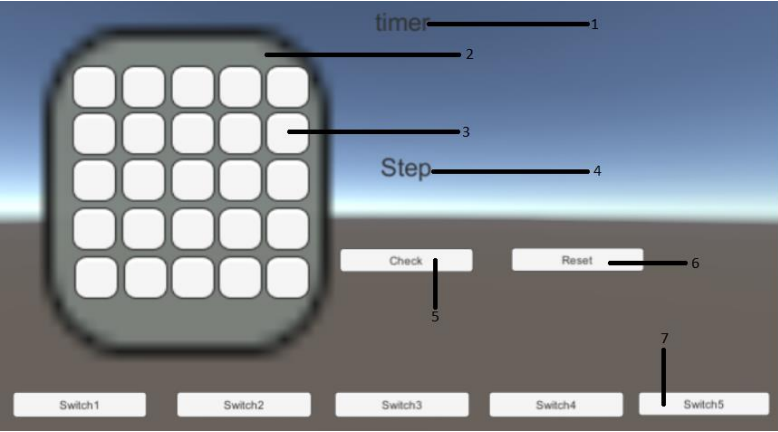
Tabel 3.17 Atribut Antarmuka Soal Pertama

No.	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan
1	<i>txtTimer</i>	<i>Text</i>	Teks untuk Hitung Mundur Waktu Soal
2	<i>GameObject</i>	<i>Game Object</i>	Wadah untuk menyimpan piringan-piringan operasi <i>stack</i> .
3	<i>Violet</i>	<i>Game Object</i>	Piringan berwarna ungu.
4	<i>Yellow</i>	<i>Game Object</i>	Piringan berwarna kuning.
5	<i>Blue</i>	<i>Game Object</i>	Piringan berwarna biru.

No.	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan
6	<i>GameObject(1)</i>	<i>Game Object</i>	Wadah untuk menyimpan piringan-piringan operasi <i>queue</i> .
7	<i>Red</i>	<i>Game Object</i>	Piringan berwarna merah.
8	<i>Gray</i>	<i>Game Object</i>	Piringan berwarna abu-abu.
9	<i>btnCheck</i>	<i>Button</i>	Tombol aksi untuk memeriksa jawaban.
10	<i>btnReset</i>	<i>Button</i>	Tombol aksi untuk mengulangi letak piringan-piringan seperti semula.
11	<i>btnPop1</i>	<i>Button</i>	Tombol aksi untuk melakukan operasi <i>pop</i> pada wadah <i>stack</i> .
12	<i>btnPop2</i>	<i>Button</i>	Tombol aksi untuk melakukan operasi <i>dequeue</i> pada wadah <i>queue</i> .
13	<i>txtStep</i>	<i>Text</i>	Teks untuk menghitung jumlah langkah yang dilakukan pengguna untuk menjawab soal.
14	<i>tutor</i>	<i>Image</i>	Gambar untuk menjelaskan kontrol tombol pada soal.
15	<i>btnSend</i>	<i>Button</i>	Tombol aksi untuk lanjut pada soal berikutnya.

3.2.5.9 Rancangan Halaman Antarmuka Soal Kedua

Pada Halaman ini akan ditampilkan uraian soal kedua. Atribut antarmuka dapat dilihat pada Tabel 3.18.



Gambar 3.30 Rancangan Antarmuka Soal Kedua

Tabel 3.18 Atribut Antarmuka Soal Kedua

No.	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan
1	<i>txtTimer</i>	<i>Text</i>	Teks untuk Hitung Mundur Waktu Soal
2	<i>GameObject</i>	<i>Game Object</i>	Wadah untuk menyimpan piringan-piringan operasi <i>stack</i> .
3	<i>Violet</i>	<i>Game Object</i>	Piringan berwarna ungu.
4	<i>Yellow</i>	<i>Game Object</i>	Piringan berwarna kuning.
5	<i>Blue</i>	<i>Game Object</i>	Piringan berwarna biru.
6	<i>GameObject(1)</i>	<i>Game Object</i>	Wadah untuk menyimpan piringan-piringan operasi <i>queue</i> .
7	<i>Red</i>	<i>Game Object</i>	Piringan berwarna merah.
8	<i>Gray</i>	<i>Game Object</i>	Piringan berwarna abu-abu.
9	<i>btnCheck</i>	<i>Button</i>	Tombol aksi untuk memeriksa jawaban.

No	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan
10	<i>btnReset</i>	<i>Button</i>	Tombol aksi untuk mengulangi letak piringan-piringan seperti semula.
11	<i>btnPop1</i>	<i>Button</i>	Tombol aksi untuk melakukan operasi <i>pop</i> pada wadah <i>stack</i> .
12	<i>btnPop2</i>	<i>Button</i>	Tombol aksi untuk melakukan operasi <i>dequeue</i> pada wadah <i>queue</i> .
13	<i>txtStep</i>	<i>Text</i>	Teks untuk menghitung jumlah langkah yang dilakukan pengguna untuk menjawab soal.
14	<i>tutor</i>	<i>Image</i>	Gambar untuk menjelaskan kontrol tombol pada soal.
15	<i>btnSend</i>	<i>Button</i>	Tombol aksi untuk lanjut pada soal berikutnya.

3.2.5.10 Rancangan Halaman Antarmuka Selesai

Pada halaman ini akan ditampilkan nilai yang didapatkan oleh pengguna. Atribut antarmuka selesai dapat dilihat pada Tabel 3.18.



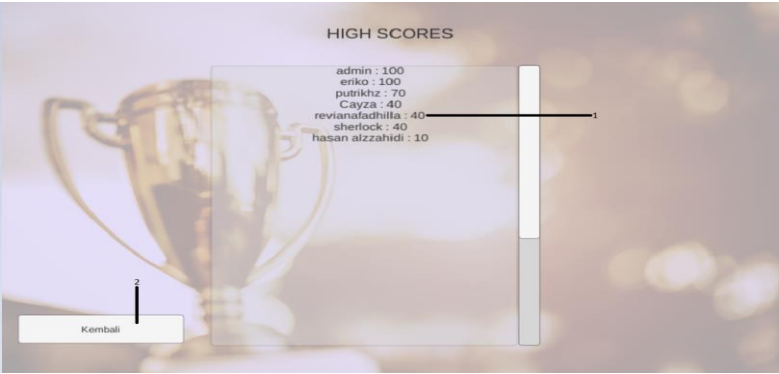
Gambar 3.31 Rancangan Antarmuka Selesai

Tabel 3.19 Atribut Antarmuka Halaman Selesai

No.	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan
1	<i>Nilai</i>	<i>Text</i>	Informasi total nilai
2	<i>Btnrating</i>	<i>Button</i>	Tombol aksi menuju halaman Ranking
3	<i>btnLogout</i>	<i>Button</i>	Tombol aksi keluar dari sistem pengerjaan soal dan menuju halaman Beranda

3.2.5.11 Rancangan Halaman Antarmuka *Ranking*

Pada halaman ini akan ditampilkan nilai dari semua peserta. Atribut antarmuka ranking dapat dilihat pada Tabel 3.19.



Gambar 3.32 Rancangan Antarmuka *Ranking*

Tabel 3.20 Atribut Antarmuka Halaman *Ranking*

No.	Nama Atribut Antarmuka	Jenis Atribut	Kegunaan
1	<i>Text</i>	<i>Text</i>	Informasi peringkat peserta berupa semua nilai peserta
2	<i>btnKembali</i>	<i>Button</i>	Tombol aksi menuju halaman Selesai

BAB IV IMPLEMENTASI

Bab ini membahas implementasi yang dilakukan berdasarkan rancangan yang telah dijabarkan pada bab sebelumnya. Sebelum penjelasan implementasi akan ditunjukkan terlebih dahulu lingkungan untuk melakukan implementasi.

Pada bagian implementasi ini juga akan dijelaskan mengenai fungsi-fungsi yang digunakan dalam program tugas akhir ini dan disertai dengan kode sumber masing-masing fungsi utama.

4.1 Lingkungan Implementasi

Spesifikasi perangkat keras serta perangkat lunak yang digunakan dalam tahap implementasi perangkat lunak tugas akhir ini.

Tabel 4.1 Lingkungan Implementasi Perangkat Lunak

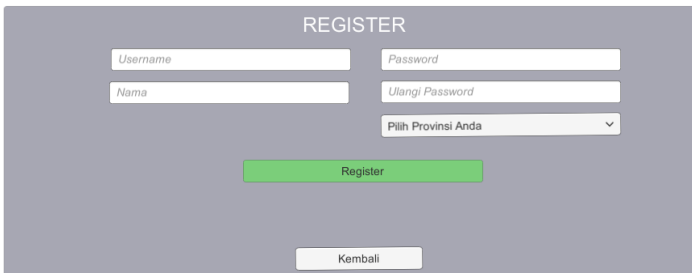
Perangkat Keras	1. Laptop MSI 2. Prosesor Intel(R) Core(TM) i7-6700HQ CPU @ 2.60GHz 3. Memori (RAM) 12.00 GB
Perangkat Lunak	1. System operasi Windows 10 (64-bit) 2. Unity 3. Visual Studio 4. Google Chrome 5. Mozilla Firefox 6. Sybase PowerDesigner 7. Microsoft Word 8. Sublime

4.2 Implementasi Antarmuka Pengguna

Implementasi antarmuka pengguna berbasis web ini menggunakan HTML5. Subbab ini akan membahas tentang implementasi antarmuka. Antarmuka yang dibuat berdasarkan rancangan antarmuka yang telah dijabarkan pada Bab 3.

4.2.1 Implementasi Antarmuka Halaman Register Pengguna

Antarmuka halaman register pada Gambar 4.1 merupakan halaman yang digunakan oleh pengguna untuk registrasi ke sistem. Terdapat tempat isian untuk *username*, nama, *password*, Ulangi *password*, *dropdown* provinsi, tombol *Register* serta tombol *Kembali*.



Gambar 4.1 Implementasi Antarmuka Halaman Register Pengguna

```

1. using System.Collections;
2. using System.Collections.Generic;
3. using UnityEngine.Networking;
4. using UnityEngine;
5. using UnityEngine.UI;
6. using System;
7.
8. public class Registrasi : MonoBehaviour {

```

Kode Sumber 4.1 Fungsi *Register* Pengguna (Bagian Pertama)

```

9.     List<string> names = new List<string>() { "Pili
h Provinsi Anda", "ACEH", "SUMATERA UTARA", "SUMATE
10.     "RAU", "JAMBI", "SUMATERA SELATAN", "BENGK
ULU", "LAMPUNG", "KEPULAUAN BANGKA BELITUNG", "KEPU
LAUAN RIAU", "DKI JAKARTA",
11.     "JAWA BARAT", "JAWA TENGAH", "DI YOGYAKARTA
", "JAWA TIMUR", "BANTEN", "BALI", "NUSA TENGGARA B
ARAT", "NUSA TENGGARA TIMUR",
12.     "KALIMANTAN BARAT", "KALIMANTAN TENGAH", "K
ALIMANTAN SELATAN", "KALIMANTAN TIMUR", "KALIMANTAN
UTARA", "SULAWESI UTARA",
13.     "SULAWESI TENGAH", "SULAWESI SELATAN", "SUL
AWESI TENGGARA", "GORONTALO", "SULAWESI BARAT", "MA
LUKU", "MALUKU UTARA",
14.     "PAPUA BARAT", "PAPUA"
15. };
16.
17. public Dropdown dropdown;
18. public Text pilihprovinsi;
19.
20. public InputField r_username;
21. public InputField r_name;
22. public InputField r_pass;
23. public InputField r_repass;
24.
25. private string username_r;
26. private string name_r;
27. private string pass_r;
28. private string repass_r;
29.
30. private string provinsi;
31. public Text error;
32. private string g_string;
33. private Color green = Color.blue;
34. private Color red = new Color(64, 0, 0, 1);
35. public void Dropdown_indexchanged(int index)
36. {
37.     pilihprovinsi.text = names [index];
38. }

```

Kode Sumber 4.2 Fungsi *Register Pengguna* (Bagian Kedua)

```

39.     void Start()
40.     {
41.         PopulateList ();
42.     }
43.     void PopulateList(){
44.         dropdown.AddOptions (names);
45.     }
46.     public void submit()
47.     {
48.         error.text = "";
49.         username_r = r_username.text;
50.         name_r = r_name.text;
51.         pass_r = r_pass.text;
52.         repass_r = r_repass.text;
53.         provinsi = pilihprovinsi.text;
54.         StartCoroutine(sign_up(username_r, name_r,
pass_r, provinsi, 5f));
55.     }
56.     IEnumerator sign_up(string elm, string elm1, st
ring elm2, string elm3, float delay)
57.     {
58.         if (pass_r == repass_r)
59.         {
60.             if (provinsi != "Pilih Provinsi Anda")
61.             {
62.                 UnityWebRequest link = UnityWebRequ
est.Get("http://somethingnotright.dx.am/registrasi.
php?r_username=" + elm + "&r_nama=" + elm1 + "&r_pa
ss="+elm2 + "&r_provinsi="+elm3);
63.                 yield return link.Send();
64.                 yield return new WaitForSeconds(del
ay);
65.                 if (link.isError)
66.                 {
67.                     Debug.Log(link.error);
68.                 }
69.                 else
70.                 {
71.                     g_string = link.downloadHandler
.text;

```

Kode Sumber 4.3 Fungsi *Register* Pengguna (Bagian Ketiga)

```

72.             show_error(g_string.ToString());
73.         }
74.     }
75.     else error.text = "Silahkan Pilih Provinsi Anda";
76. }
77.     else error.text = "Password Tidak Cocok";
78. }
79.
80. private void show_error(string elm)
81. {
82.     string[] arr = null;
83.     arr = elm.Split(new string[] { " " }, StringSplitOptions.RemoveEmptyEntries);
84.     if (arr[3] == "Berhasil") {
85.         error.color = green;
86.         error.text = elm;
87.     }
88.     else
89.     {
90.         error.color = red;
91.         error.text = elm;
92.     }
93. }
94. }

```

Kode Sumber 4.4 Fungsi *Register* Pengguna (Bagian Keempat)

```

1. <?php
2.
3. header("Access-Control-Allow-Credentials: true");
4. header('Access-Control-Allow-Origin: *');
5. header('Access-Control-Allow-Methods: POST, GET, OPTIONS');
6. header('Access-Control-Allow-Headers: Accept, X-Access-Token, X-Application-Name, X-Request-Sent-Time');

```

Kode Sumber 4.5 Kontrol *Register* Pengguna (Bagian Pertama)

```

7. error_reporting(E_ALL ^ E_DEPRECATED);
8.
9. $Server = "fdb17.awardspace.net";
10. $User = "2326813_admin";
11. $Pass = "zzz1312000";
12. $Db = "2326813_admin";
13. $Connection = mysql_connect($Server, $User, $Pass,
    $Db);
14.
15. if(mysql_error()){
16.     die("Connection Failed!=". mysql_error());
17. }
18. mysql_select_db("2326813_admin", $Connection) or di
    e("Could not load to database" . mysql_error());
19.
20. $username = $_GET["r_username"];
21. $nama = $_GET["r_nama"];
22. $pass = $_GET["r_pass"];
23. $prov = $_GET["r_provinsi"];
24.
25. $check = mysql_query("SELECT * FROM peserta WHERE `
    username` = '". $username.'");
26. $numrows = mysql_num_rows($check);
27.
28. if((emptyempty($username))||(emptyempty($nama))||(e
    mptyempty($pass))||(emptyempty($prov))){
29.     echo "Data yang Anda masukkan kurang";
30.     die;
31. }
32. elseif($numrows==0)
33. {
34.     $sql = mysql_query("INSERT INTO peserta (id, us
        ername, name, password, provinsi) VALUES ('', '". $u
        sername.', '."'.'$nama.', '."'.'md5(md5($pass))."', '."'.'
        $prov.'")");
35.     $soal0 = mysql_query("INSERT INTO jawab (id_jaw
        ab, id, id_soal) VALUES (CONCAT(LAST_INSERT_ID()), 'S
        0'), LAST_INSERT_ID(), 'S0')");

```

Kode Sumber 4.6 Kontrol Register Pengguna (Bagian Kedua)


```

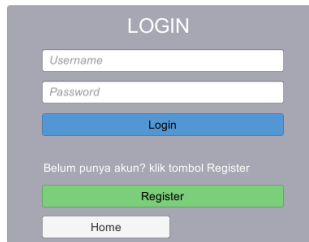
36.     $soal1 = mysql_query("INSERT INTO jawab (id_jaw
    ab, id, id_soal) VALUES (CONCAT(LAST_INSERT_ID(), 'S
    1'), LAST_INSERT_ID(), 'S1')");
37.     $soal2 = mysql_query("INSERT INTO jawab (id_jaw
    ab, id, id_soal) VALUES (CONCAT(LAST_INSERT_ID(), 'S
    2'), LAST_INSERT_ID(), 'S2')");
38.     $soal3 = mysql_query("INSERT INTO jawab (id_jaw
    ab, id, id_soal) VALUES (CONCAT(LAST_INSERT_ID(), 'S
    3'), LAST_INSERT_ID(), 'S3')");
39.
40.
41.     if($sql){
42.         echo "memasukkan username ".$username." Ber
    hasil";
43.     }
44.     else{
45.         die("memasukkan username ".$username." Gaga
    l");
46.     }
47. }
48. else
49. {
50.     die("Username sudah ada !");
51. }
52.
53. mysql_close($Connection);
54.
55. ?>

```

Kode Sumber 4.7 Kontrol *Register* Pengguna (Bagian Ketiga)

4.2.2 Implementasi Antarmuka Halaman Login Pengguna

Antarmuka halaman login pada Gambar 4.1 merupakan halaman yang digunakan oleh pengguna untuk masuk ke sistem dan mengakses halaman soal. Terdapat tempat isian untuk *username*, *password*, tombol *Login* tombol *Register* serta tombol *Kembali*.



Gambar 4.2 Implementasi Antarmuka Halaman Login Pengguna

```

1. using System.Collections;
2. using System.Collections.Generic;
3. using UnityEngine;
4. using UnityEngine.UI;
5. using System;
6. using UnityEngine.Networking;
7.
8.
9. public class Login : MonoBehaviour {
10.
11.     public Button submitBtn;
12.
13.     public InputField r_username;
14.     public InputField r_pass;
15.
16.     private string username_r;
17.     private string pass_r;
18.
19.     public Text error;
20.
21.     private string g_string;
22.     private Color green = Color.blue;
23.     private Color red = new Color(64, 0, 0, 1);
24.
25.     void Start(){

```

Kode Sumber 4.8 Fungsi *Login* Pengguna (Bagian Pertama)

```

26.         Button btn = submitBtn.GetComponent<Button>
           ();
27.         btn.onClick.AddListener(Submit);
28.     }
29.
30.     public void Submit()
31.     {
32.         error.text = "";
33.         username_r = r_username.text;
34.         pass_r = r_pass.text;
35.
36.         //error.text = username_r + pass_r;
37.         StartCoroutine(log_in(username_r, pass_r));
38.
39.         if (error.color == green)
40.         {
41.             PlayerPrefs.SetString("username", usern
ame_r);
42.             //Debug.Log(PlayerPrefs.GetString("user
name").ToString());
43.             Application.LoadLevel("pagePilihNomor")
;
44.         }
45.         else return;
46.     }
47.     IEnumerator log_in(string el, string el1)
48.     {
49.         UnityWebRequest link = UnityWebRequest.Get(
"http://somethingnotright.dx.am/login2.php?r_userna
me=" + el + "&r_pass=" + el1);
50.         yield return link.Send();
51.         if (link.isError)
52.         {
53.             Debug.Log(link.error);
54.         }
55.         else
56.         {
57.             g_string = link.downloadHandler.text;
58.

```

Kode Sumber 4.9 Fungsi *Login* Pengguna (Bagian Kedua)

```

59.         show_error(g_string.ToString());
60.     }
61. }
62. private void show_error(string el)
63. {
64.     string[] arr = null;
65.     arr = el.Split(new string[] { " " }, String
SplitOptions.RemoveEmptyEntries);
66.     if (arr[2] == "Berhasil")
67.     {
68.         error.color = green;
69.         error.text = el;
70.     }
71.     else
72.     {
73.         error.color = red;
74.         error.text = el;
75.     }
76. }
77. }

```

Kode Sumber 4.10 Fungsi *Login* Pengguna (Bagian Ketiga)

```

1.  <?php
2.
3.  header("Access-Control-Allow-Credentials: true");
4.  header('Access-Control-Allow-Origin: *');
5.  header('Access-Control-Allow-
Methods: POST, GET, OPTIONS');
6.  header('Access-Control-Allow-Headers: Accept, X-
Access-Token, X-Application-Name, X-Request-Sent-
Time');
7.  error_reporting(E_ALL ^ E_DEPRECATED);
8.  $Server = "fdb17.awardspace.net";
9.  $User = "2326813_admin";
10. $Pass = "zzz1312000";
11. $Db = "2326813_admin";
12. $Connection = mysql_connect($Server, $User, $Pass,
$Db);

```

Kode Sumber 4.11 Kontrol *Login* Pengguna (Bagian Pertama)

```

13. if(mysql_error()){
14.     die("Connection Failed!=". mysql_error());
15. }
16. mysql_select_db("2326813_admin", $Connection) or die
    e("Could not load to database" . mysql_error());
17.
18. $username = $_GET["r_username"];
19. $pass = md5(md5($_GET["r_pass"]));
20.
21. $check = mysql_query("SELECT * FROM peserta WHERE `
    username` = '". $username.'");
22. $numrows = mysql_num_rows($check);
23.
24. if((empty($username))||(empty($pass))){
25.     echo "Data yang Anda masukkan kurang";
26. }
27.
28. else if($numrows == 0)
29. {
30.     die("Username yang Anda masukkan salah");
31. }
32. else
33. {
34.     while($row = mysql_fetch_assoc($check))
35.     {
36.         if($pass == $row['password'])
37.         {
38.             die("Login Anda Berhasil");
39.         }
40.         else
41.         {
42.             die("Password yang anda masukkan salah"
43. );
44.         }
45.     }
46.
47. ?>

```

Kode Sumber 4.12 Kontrol Login Pengguna (Bagian Kedua)

4.2.3 Implementasi Antarmuka Halaman Beranda

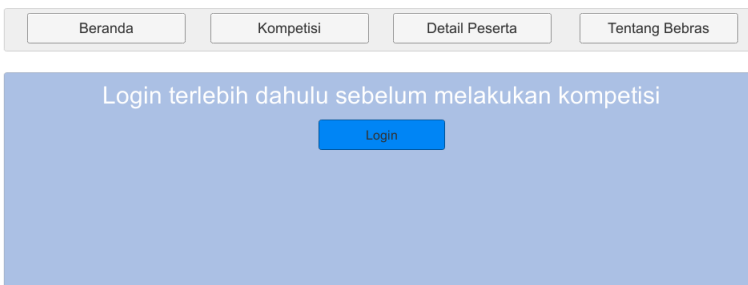
Antarmuka halaman beranda pada Gambar 4.1 merupakan halaman yang akan muncul pertama kali saat membuka *web*.



Gambar 4.3 Implementasi Halaman Antarmuka Beranda

4.2.4 Implementasi Antarmuka Halaman Kompetisi

Antarmuka Halaman kompetisi merupakan halaman untuk masuk kompetisi melalui *login* terlebih dahulu.



Gambar 4.4 Implementasi Halaman Antarmuka Kompetisi

4.2.5 Implementasi Antarmuka Halaman Detail Peserta

Antarmuka halaman detail peserta merupakan halaman untuk melihat data peserta.



Gambar 4.5 Implementasi Halaman Antarmuka Detail Peserta

```

1. using System.Collections;
2. using System.Collections.Generic;
3. using UnityEngine;
4. using UnityEngine.UI;
5.
6. public class DetailPeserta : MonoBehaviour {
7.
8.     public Text data;
9.     public Text ttl;
10.    public Text provin;
11.
12.    IEnumerator Start(){
13.        WWW itemsdata = new WWW("http://somethingno
    tright.dx.am/detailpeserta.php");
14.        WWW total = new WWW("http://somethingnotrig
    ht.dx.am/totalpeserta.php");

```

Kode Sumber 4.13 Fungsi Detail Peserta (Bagian Pertama)

```

15.         WWW prov = new WWW("http://somethingnotrigh
    t.dx.am/detailprovinsi.php");
16.         yield return itemsdata;
17.         yield return total;
18.         yield return prov;
19.         string itemsdatastring = itemsdata.text;
20.         string itemstotal = total.text;
21.         string itemsprov = prov.text;
22.         print (itemsdatastring);
23.         print (itemstotal);
24.         print (itemsprov);
25.         data.text = itemsdatastring;
26.         ttl.text = itemstotal;
27.         provin.text = itemsprov;
28.     }
29. }

```

Kode Sumber 4.14 Fungsi Detail Peserta (Bagian Kedua)

```

1. <?php
2.
3. header("Access-Control-Allow-Credentials: true");
4. header('Access-Control-Allow-Origin: *');
5. header('Access-Control-Allow-
    Methods: POST, GET, OPTIONS');
6. header('Access-Control-Allow-Headers: Accept, X-
    Access-Token, X-Application-Name, X-Request-Sent-
    Time');
7.
8. error_reporting(E_ALL ^ E_DEPRECATED);
9.
10. $Server = "fdb17.awardspace.net";
11. $User = "2326813_admin";
12. $Pass = "zzz1312000";
13. $Db = "2326813_admin";
14. $Connection = mysql_connect($Server, $User, $Pass,
    $Db);
15.

```

Kode Sumber 4.15 Kontrol Detail Peserta (Bagian Pertama)


```

16. if(mysql_error()){
17.     die("Connection Failed!=" . mysql_error());
18. }
19. mysql_select_db("2326813_admin", $Connection) or die
    e("Could not load to database" . mysql_error());
20.
21. $check = "SELECT username, name, provinsi FROM peser
    ta";
22. $result=mysql_query($check);
23. $numrows = mysql_num_rows($result);
24.
25. //while($row=mysql_fetch_array($result))
26. if($numrows > 0 ){
27.     while($row=mysql_fetch_array($result))
28.     {
29.         echo "\nUsername : ".$row['username']." ";
30.
31.         echo "(".$row['name'].") ";
32.         echo "- ".$row['provinsi']."\n";
33.     }
34. }>

```

Kode Sumber 4.16 Kontrol Detail Peserta (Bagian Kedua)

```

1. <?php
2.
3. header("Access-Control-Allow-Credentials: true");
4. header('Access-Control-Allow-Origin: *');
5. header('Access-Control-Allow-
    Methods: POST, GET, OPTIONS');
6. header('Access-Control-Allow-Headers: Accept, X-
    Access-Token, X-Application-Name, X-Request-Sent-
    Time');
7.
8. error_reporting(E_ALL ^ E_DEPRECATED);
9.
10. $Server = "fdb17.awardspace.net";
11. $User = "2326813_admin";

```

Kode Sumber 4.17 Kontrol Jumlah Peserta (Bagian Pertama)

```

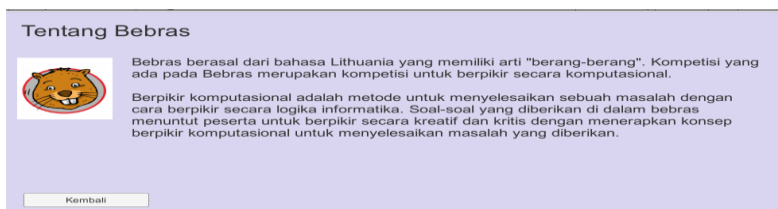
12. $Pass = "zzz1312000";
13. $Db = "2326813_admin";
14. $Connection = mysql_connect($Server, $User, $Pass,
    $Db);
15.
16. if(mysql_error()){
17.     die("Connection Failed!=". mysql_error());
18. }
19. mysql_select_db("2326813_admin", $Connection) or die("Could not load to database" . mysql_error());
20.
21. $check = "SELECT COUNT(username) AS jumlah FROM peserta";
22. $result=mysql_query($check);
23. $numrows = mysql_num_rows($result);
24. if($numrows > 0 ){
25.     while($row=mysql_fetch_array($result))
26.     {
27.         echo "Jumlah Peserta : ".$row['jumlah']."";
28.     }
29. }
30. ?>

```

Kode Sumber 4.18 Kontrol Jumlah Peserta (Bagian Kedua)

4.2.6 Implementasi Halaman Antarmuka Tentang Bebras

Antarmuka halaman tentang bebras merupakan halaman untuk melihat informasi tentang bebras.



Gambar 4.6 Implementasi Halaman Antarmuka Tentang Bebras

4.2.7 Implementasi Antarmuka Halaman Penjelasan Soal

Antarmuka halaman penjelasan soal merupakan halaman untuk melihat penjelasan soal.

4.2.8 Implementasi Antarmuka Halaman Selesai

Antarmuka halaman soal merupakan halaman yang ditampilkan setelah selesai mengerjakan soal, berisi nilai yang didapatkan oleh peserta.



Gambar 4.7 Implementasi Antarmuka Halaman Selesai

```

1. using System.Collections;
2. using System.Collections.Generic;
3. using UnityEngine;
4. using UnityEngine.UI;
5. using System;
6. using UnityEngine.Networking;
7.
8.
9. public class Selesai : MonoBehaviour {
10.
11.     public Text nilai;

```

Kode Sumber 4.19 Fungsi Selesai (Bagian Pertama)

```

12.     public Button submitBtn;
13.     string user;
14.
15.
16.     // Use this for initialization
17.     void Start()
18.     {
19.         Debug.Log(PlayerPrefs.GetString("username")
20.         .ToString());
21.         int score0 = PlayerPrefs.GetInt("scoreno0")
22.         ;
23.         int score1 = PlayerPrefs.GetInt("scoreno1")
24.         ;
25.         int score2 = PlayerPrefs.GetInt("scoreno2")
26.         ;
27.         int score3 = PlayerPrefs.GetInt("scoreno3")
28.         ;
29.         user = PlayerPrefs.GetString("username").To
30.         String();
31.         int total = score0 + score1 + score2 + scor
32.         e3;
33.         print(total);
34.         PlayerPrefs.SetInt("TampilNilai", total);
35.         StartCoroutine(Postscore(user, total));
36.         Debug.Log(PlayerPrefs.GetInt("TampilNilai")
37.         );
38.         string Nilai = PlayerPrefs.GetInt("TampilNi
39.         lai").ToString();
40.         nilai.text = Nilai;
41.         Button btn = submitBtn.GetComponent<Button>
42.         ();
43.         btn.onClick.AddListener(Submit);
44.     }
45.
46.     public void Submit(){
47.         PlayerPrefs.DeleteAll();
48.         Application.LoadLevel ("pageHome");
49.     }

```

Kode Sumber 4.20 Fungsi Selesai (Bagian Kedua)

```

40.     IEnumerator Postscore(string user, int total)
41.     {
42.         UnityWebRequest link = UnityWebRequest.Get(
43.             "http://somethingnotright.dx.am/selesai.php?r_usern
44.             ame=" + user + "&r_nilai=" + total);
45.         yield return link.Send();
46.         if (link.isError)
47.         {
48.             Debug.Log(link.error);
49.         }
50.         // Update is called once per frame
51.         void Update () {
52.             }
53.     }

```

Kode Sumber 4.21 Fungsi Selesai (Bagian Ketiga)

```

1.  <?php
2.
3.  header("Access-Control-Allow-Credentials: true");
4.  header('Access-Control-Allow-Origin: *');
5.  header('Access-Control-Allow-
6.  Methods: POST, GET, OPTIONS');
7.  header('Access-Control-Allow-Headers: Accept, X-
8.  Access-Token, X-Application-Name, X-Request-Sent-
9.  Time');
10. error_reporting(E_ALL ^ E_DEPRECATED);
11.
12. $Server = "fdb17.awardspace.net";
13. $User = "2326813_admin";
14. $Pass = "zzz1312000";
15. $Db = "2326813_admin";
16. $Connection = mysql_connect($Server, $User, $Pass,
17.     $Db);

```

Kode Sumber 4.22 Kontrol Fungsi Selesai (Bagian Pertama)

```

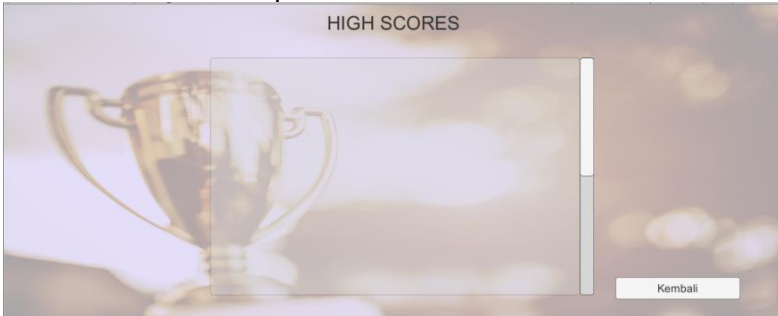
15. if(mysql_error()){
16.     die("Connection Failed!=". mysql_error());
17. }
18. mysql_select_db("2326813_admin", $Connection) or die
    e("Could not load to database" . mysql_error());
19.
20. $username = $_GET["r_username"];
21. $nilai = $_GET["r_nilai"];
22.
23. $check = mysql_query("SELECT * FROM peserta WHERE `
    username` = '". $username.'");
24. $numrows = mysql_num_rows($check);
25.
26. if($numrows == 0)
27. {
28.     die("Error");
29. }
30. else
31. {
32.     while($row = mysql_fetch_assoc($check))
33.     {
34.         if($username == $row['username'])
35.         {
36.             $sql = mysql_query("UPDATE `peserta` SE
    T `nilai` = '". $nilai.'" WHERE `username` = '". $use
    rname.'"");
37.         }
38.         else
39.         {
40.             die("error");
41.         }
42.     }
43. }
44.
45. ?>

```

Kode Sumber 4.23 Kontrol Fungsi Selesai (Bagian Kedua)

4.2.9 Implementasi Antarmuka Halaman *Ranking*

Antarmuka halaman *ranking* merupakan halaman yang berisi data semua nilai peserta.



Gambar 4.7 Implementasi Antarmuka Halaman Ranking

```

1. using System.Collections;
2. using System.Collections.Generic;
3. using UnityEngine;
4. using UnityEngine.UI;
5.
6. public class Ranking : MonoBehaviour {
7.
8.     public Text data;
9.
10.    IEnumerator Start(){
11.        WWW itemsdata = new WWW("http://somethingno
    tright.dx.am/ranking.php");
12.        yield return itemsdata;
13.        string itemsdatastring = itemsdata.text;
14.        print (itemsdatastring);
15.        data.text = itemsdatastring;
16.    }
17. }

```

Kode Sumber 4.24 Fungsi *Ranking*

```

1. <?php
2.
3. header("Access-Control-Allow-Credentials: true");
4. header('Access-Control-Allow-Origin: *');
5. header('Access-Control-Allow-
  Methods: POST, GET, OPTIONS');
6. header('Access-Control-Allow-Headers: Accept, X-
  Access-Token, X-Application-Name, X-Request-Sent-
  Time');
7.
8. error_reporting(E_ALL ^ E_DEPRECATED);
9.
10. $Server = "fdb17.awardspace.net";
11. $User = "2326813_admin";
12. $Pass = "zzz1312000";
13. $Db = "2326813_admin";
14. $Connection = mysql_connect($Server, $User, $Pass,
  $Db);
15.
16. if(mysql_error()){
17.     die("Connection Failed!=". mysql_error());
18. }
19. mysql_select_db("2326813_admin", $Connection) or di
  e("Could not load to database" . mysql_error());
20.
21. $check = "SELECT username, nilai FROM peserta ORDER
  BY nilai DESC";
22. $result=mysql_query($check);
23. $numrows = mysql_num_rows($result);
24.
25. if($numrows > 0 ){
26.     while($row=mysql_fetch_array($result))
27.     {
28.         echo " " . $row['username']. " : " . $row['ni
  lai']. "\n" ;
29.     }
30. }
31.
32. ?>

```

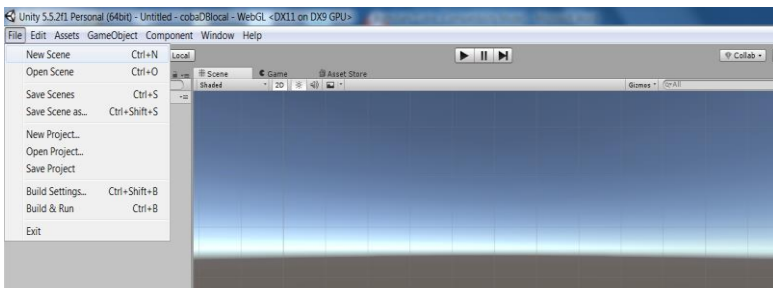
Kode Sumber 4.25 Kontrol Fungsi *Ranking*

4.3 Implementasi Pembuatan Aplikasi

Pada tahap implementasi pembuatan aplikasi ini akan membahas tentang implementasi pembuatan *scene*, implementasi import *assets*, implementasi *load* objek ke *scene*, serta implementasi pembuatan *script*.

4.3.1 Implementasi Pembuatan *Scene*

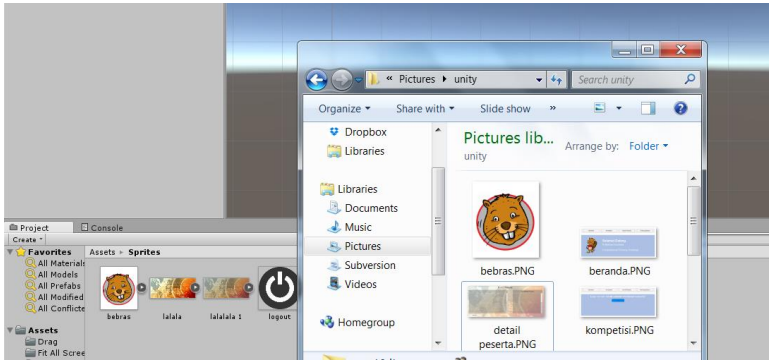
Untuk menambahkan *scene*, hal yang perlu dilakukan adalah menuju menu “File” lalu pilih “New Scene”. Setelah *scene* baru terbuat, kita dapat menambah objek kedalam *scene* dan mengatur sesuai rancangan yang dibuat.



Gambar 4.8 Pembuatan *Scene*

4.3.2 Implementasi *Import Asset*

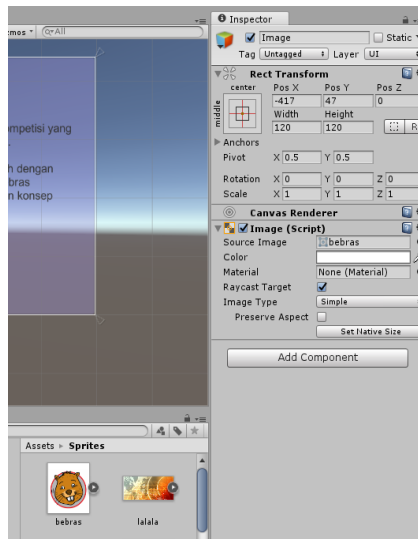
Implementasi *import assets* dapat dilakukan dengan cara *drag and drop* dari file explore ke tab “Project” didalam folder “Assets” pada unity. Kita juga dapat mengatur pengelompokan *asset* dengan membuat folder-folder supaya lebih rapi dan memudahkan dalam pencarian *asset*.



Gambar 4.9 Implementasi *Import Asset*

4.3.3 Implementasi *Load Objek ke Scene*

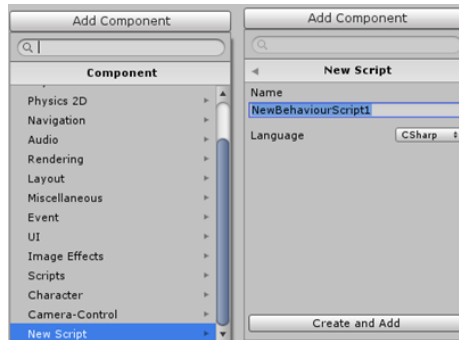
Untuk memasukan objek-objek ke dalam *scene* dapat dimasukkan dengan cara *drag and drop* berkas dari tab “Project” ke dalam *Scene*.



Gambar 4.10 Implementasi *Load* Objek ke *Scene*

4.3.4 Implementasi Pembuatan *Script*

Untuk membuat *script* yang mempengaruhi objek adalah dengan memilih terlebih dahulu objek yang akan diberi *script*. Pada tab “Inspector” tambahkan komponen *script*. Jika tidak ada *script* yang sesuai, kita akan membuat file *script* baru dan mengubah isi file sesuai yang kita inginkan.



Gambar 4.11 Implementasi Pembuatan Script

4.3.5 Implementasi Pembuatan Soal

4.3.5.1 Implementasi Pembuatan Soal Pertama

Soal pertama mengimplementasikan fungsi *stack* dan *queue* pada keahlian informatika. Terdapat dua tempat untuk menyimpan piringan-piringan, kotak pertama mengimplementasikan *stack*, yaitu *Last In First Out* (LIFO) dan kotak kedua mengimplementasikan *queue*, yaitu *First In First Out* (FIFO).

```

1. public Stack<GameObject> Deck1 = new Stack<GameObje
   ct>();
2.     public GameObject stack;
3.     public List<GameObject> stackChild;
4.     public Button pop1;
5.     public float positionRangeY;
6.     public QueueScript2 queue;
7.     public float BasePosition;

```

Kode Sumber 4.26 Inisiasi Variabel Stack

```

1. public Queue<GameObject> Deck2 = new Queue<GameObje
   ct>();
2.     public GameObject queue;
3.     public List<GameObject> queueChild;
4.     public Button pop2;
5.     public StackScript stack;
6.     public float PositionRangeY;
7.     public float BasePosition;

```

Kode Sumber 4.27 Inisiasi Variabel *Queue*

```

1. public void popA()
2. {
3.     CheckStack.stepCount++;
4.     GameObject go = Deck1.Pop();
5.     stackChild.Remove(go);
6.     queue.pushB(go);
7. }

```

Kode Sumber 4.28 *Pop* dari *Stack*

```

1. public void pushA(GameObject go)
2. {
3.     Vector3 go2 = new Vector3 (0,0,-1);
4.     if (Deck1.Count != 0)
5.         go2 = Deck1.Peek().transform.localPosition
6.     else
7.         go2 = new Vector3(0, BasePosition, -1);
8.     stackChild.Add(go);
9.     Deck1.Push(go);
10.    Vector3 oldPosition = go.transform.localPos
   ition;
11.    go.transform.parent = stack.transform;
12.    go.transform.localPosition = new Vector3(ol
   dPosition.x, go2.y + positionRangeY, go.transform.l
   ocalPosition.z);
13. }

```

Kode Sumber 4.29 *Push* menuju *Stack*

```

1. public void popB()
2. {
3.     CheckStack.stepCount++;
4.     GameObject go = Deck2.Dequeue();
5.     queueChild.Remove(go);
6.     stack.pushA(go);
7.     ResetPosition();
8. }

```

Kode Sumber 4.30 Dequeue dari queue

```

1. public void ResetPosition()
2. {
3.     for(int i = 0; i < queueChild.Count; i++)
4.     {
5.         queueChild[i].transform.localPosition =
           new Vector3(queueChild[i].transform.localPosition.
           x, queueChild[i].transform.localPosition.y -
           PositionRangeY, queueChild[i].transform.localPosit
           ion.z);
6.     }
7. }

```

Kode Sumber 4.31 Reset Posisi Queue

```

1. public void pushB(GameObject go)
2. {
3.     Vector3 go2 = new Vector3(0, 0, -1);
4.     if (queueChild.Count != 0)
5.         go2 = queueChild[queueChild.Count -
           1].transform.localPosition;
6.     else
7.         go2 = new Vector3(0, BasePosition, -
           1);
8.     Deck2.Enqueue(go);
9.     queueChild.Add(go);

```

Kode Sumber 4.32 Enqueue Menuju Queue (Bagian Pertama

```

10.         Vector3 oldPosition = go.transform.localPosition
           ition;
11.         go.transform.parent = queue.transform;
12.         go.transform.localPosition = new Vector3(oldPosition.x, go2.y + PositionRangeY, -1);
13.     }

```

Kode Sumber 4.33 *Enqueue Menuju Queue* (Bagian Kedua)

```

1.  using System.Collections;
2.  using System.Collections.Generic;
3.  using UnityEngine;
4.  using UnityEngine.UI;
5.
6.  public class CheckStack : MonoBehaviour {
7.
8.      public Button Check;
9.      public Text Benar, Step;
10.     public static int stepCount;
11.     public QueueScript2 queue;
12.     public StackScript stack;
13.     public GameObject blue, yellow, violet, gray, red;
14.
15.     // Use this for initialization
16.     void Start () {
17.         stepCount = 0;
18.         Button btn = Check.GetComponent<Button>();
19.
20.         btn.onClick.AddListener(checkBenar);
21.     }
22.     // Update is called once per frame
23.     void Update () {
24.         Step.text = stepCount.ToString();
25.     }
26.
27.     public void checkBenar()

```

Kode Sumber 4.34 Kode Sumber Cek Jawaban (Bagian Pertama)

```

28.     {
29.         if(stack.stackChild[0]==blue && stack.stack
    Child[1]==red && stack.stackChild[2]==violet &&
30.         queue.queueChild[0]==gray && queue.queu
    eChild[1] == yellow && stepCount==6)
31.         {
32.             Benar.text = "Benar";
33.         }
34.         else
35.         {
36.             Benar.text = "Salah";
37.         }
38.     }
39.
40.     public void Reset()
41.     {
42.         stack.Reset(blue, yellow, violet);
43.         queue.Reset(gray, red);
44.         stepCount = 0;
45.     }
46. }

```

Kode Sumber 4.35 Kode Sumber Cek Jawaban (Bagian Kedua)

4.3.5.2 Implementasi Pembuatan Soal Kedua

Soal kedua mengimplementasikan fungsi matriks dan operasi matriks pada keahlian informatika. Terdapat 25 lampu pada sebuah bangunan. Pengguna diminta untuk menyalakan lampu sesuai dengan perintah. Terdapat 5 tombol *switch* dimana setiap *switch* mempunyai kombinasi yang berbeda-beda.

```

1. public bool nyala;
2. private SpriteRenderer lampu;

```

Kode Sumber 4.36 Inisiasi Variabel Status Lampu


```

1. public LampStats[] lampu;
2. public Button btn1, btn2, btn3, btn4, btn5;

```

Kode Sumber 4.37 Inisiasi Variabel Tombol dan Lampu

```

1. public void Switch1()
2. {
3.     lampu[0].nyala = !lampu[0].nyala;
4.     lampu[1].nyala = !lampu[1].nyala;
5.     lampu[3].nyala = !lampu[3].nyala;
6.     lampu[4].nyala = !lampu[4].nyala;
7.     lampu[5].nyala = !lampu[5].nyala;
8.     lampu[9].nyala = !lampu[9].nyala;
9.     lampu[15].nyala = !lampu[15].nyala;
10.    lampu[19].nyala = !lampu[19].nyala;
11.    lampu[20].nyala = !lampu[20].nyala;
12.    lampu[21].nyala = !lampu[21].nyala;
13.    lampu[23].nyala = !lampu[23].nyala;
14.    lampu[24].nyala = !lampu[24].nyala;
15. }

```

Kode Sumber 4.38 Fungsi *Switch* Pertama

```

1. public void Switch2()
2. {
3.     lampu[0].nyala = !lampu[0].nyala;
4.     lampu[1].nyala = !lampu[1].nyala;
5.     lampu[2].nyala = !lampu[2].nyala;
6.     lampu[3].nyala = !lampu[3].nyala;
7.     lampu[4].nyala = !lampu[4].nyala;
8.     lampu[5].nyala = !lampu[5].nyala;
9.     lampu[10].nyala = !lampu[10].nyala;
10.    lampu[15].nyala = !lampu[15].nyala;
11. }

```

Kode Sumber 4.39 Fungsi *Switch* Kedua

```

1.  public void Switch3()
2.      {
3.          lampu[2].nyala = !lampu[2].nyala;
4.          lampu[7].nyala = !lampu[7].nyala;
5.          lampu[10].nyala = !lampu[10].nyala;
6.          lampu[11].nyala = !lampu[11].nyala;
7.          lampu[12].nyala = !lampu[12].nyala;
8.          lampu[13].nyala = !lampu[13].nyala;
9.          lampu[14].nyala = !lampu[14].nyala;
10.         lampu[17].nyala = !lampu[17].nyala;
11.         lampu[22].nyala = !lampu[22].nyala;
12.     }

```

Kode Sumber 4.40 Fungsi *Switch* Ketiga

```

1.  public void Switch4()
2.      {
3.          lampu[0].nyala = !lampu[0].nyala;
4.          lampu[5].nyala = !lampu[5].nyala;
5.          lampu[10].nyala = !lampu[10].nyala;
6.          lampu[15].nyala = !lampu[15].nyala;
7.          lampu[20].nyala = !lampu[20].nyala;
8.          lampu[21].nyala = !lampu[21].nyala;
9.          lampu[22].nyala = !lampu[22].nyala;
10.         lampu[23].nyala = !lampu[23].nyala;
11.         lampu[24].nyala = !lampu[24].nyala;
12.     }

```

Kode Sumber 4.41 Fungsi *Switch* Keempat

```

1.  public void Switch5()
2.      {
3.          lampu[1].nyala = !lampu[1].nyala;
4.          lampu[2].nyala = !lampu[2].nyala;

```

Kode Sumber 4.42 Fungsi *Switch* Kelima (Bagian Pertama)

```

5.         lampu[3].nyala = !lampu[3].nyala;
6.         lampu[5].nyala = !lampu[5].nyala;
7.         lampu[9].nyala = !lampu[9].nyala;
8.         lampu[10].nyala = !lampu[10].nyala;
9.         lampu[14].nyala = !lampu[14].nyala;
10.        lampu[15].nyala = !lampu[15].nyala;
11.        lampu[19].nyala = !lampu[19].nyala;
12.        lampu[21].nyala = !lampu[21].nyala;
13.        lampu[22].nyala = !lampu[22].nyala;
14.        lampu[23].nyala = !lampu[23].nyala;
15.    }

```

Kode Sumber 4.43 Fungsi *Switch* Kelima (Bagian Kedua)

4.3.6 Implementasi *Random* Soal dan *Insert* Nilai

Pada saat peserta mengerjakan soal, sistem akan mengacak soal. Terdapat empat soal yang akan diacak oleh sistem. Setiap peserta selesai mengerjakan soal, maka sistem akan memasukkan nilai peserta pada soal tersebut ke dalam *database*.

```

1.  if (jawaban.text == "Benar")
2.      {
3.          PlayerPrefs.SetInt("scoreno1", 25);
4.      }
5.      else
6.      {
7.          PlayerPrefs.SetInt("scoreno1", -5);
8.      }
9.      int cekno1 = PlayerPrefs.GetInt ("scoreno1"
10. );
11.      int cekno2 = PlayerPrefs.GetInt ("scoreno2"
12. );
13.      int cekno0 = PlayerPrefs.GetInt ("scoreno0"
14. );

```

Kode Sumber 4.44 Implementasi *Random* Soal (Bagian Pertama)

```

12.         int cekno3 = PlayerPrefs.GetInt ("scoreno3"
13.     );
14.         score1 = PlayerPrefs.GetInt("scoreno1");
15.         Debug.Log (PlayerPrefs.GetInt ("scoreno1"))
16.     ;
17.         user = PlayerPrefs.GetString("username").To
18.     String();
19.         StartCoroutine(Postscore(user,idsoal, score
20.     1, 5f));
21.
22.         string acak = scene.ToString();
23.         if (cekno1 != 0 && cekno2 != 0 && cekno0 !=
24.     0 && cekno3 != 0) {
25.             Application.LoadLevel ("pageSelesai");
26.         }
27.         else if (cekno1 != 0 && cekno2 == 0 && cekn
28.     o0 == 0 && cekno3 == 0) {
29.             scene = number [UnityEngine.Random.Rang
30.     e (min, max)];
31.             acak = scene.ToString ();
32.             Debug.Log (acak);
33.             Application.LoadLevel (acak);
34.         } else if (cekno1 != 0 && cekno2 == 0 && ce
35.     kno0 != 0 && cekno3 == 0) {
36.             scene = number1 [UnityEngine.Random.Ran
37.     ge (min1, max1)];
38.             print (scene);
39.             acak = scene.ToString ();
40.             Debug.Log (acak);
41.             Application.LoadLevel (acak);
42.         } else if (cekno1 != 0 && cekno2 != 0 && ce
43.     kno0 == 0 && cekno3 == 0) {
44.             scene = number2 [UnityEngine.Random.Ran
45.     ge (min1, max1)];
46.             print (scene);
47.             acak = scene.ToString ();
48.             Debug.Log (acak);

```

Kode Sumber 4.45 Implementasi *Random* Soal (Bagian Kedua)

```

39.         Application.LoadLevel (acak);
40.     } else if (cekno1 != 0 && cekno2 == 0 && ce
    kno0 == 0 && cekno3 != 0) {
41.         scene = number3 [UnityEngine.Random.Ran
    ge (min1, max1)];
42.         print (scene);
43.         acak = scene.ToString ();
44.         Debug.Log (acak);
45.         Application.LoadLevel (acak);
46.     } else if (cekno0 == 0 && cekno2 != 0 && ce
    kno1 != 0 && cekno3 != 0 ) {
47.         Application.LoadLevel ("0");
48.     } else if (cekno0 != 0 && cekno2 == 0 && ce
    kno1 != 0 && cekno3 != 0) {
49.         Application.LoadLevel ("2");
50.     } else if (cekno0 != 0 && cekno2 != 0 && ce
    kno1 != 0 && cekno3 == 0) {
51.         Application.LoadLevel ("3");
52.     }

```

Kode Sumber 4.46 Implementasi *Random* Soal (Bagian Ketiga)

```

1. IEnumerator Postscore(string user, string idsoal, i
    nt nilai1, float delay)
2.     {
3.         UnityWebRequest link = UnityWebRequest.Get
    ("http://somethingnotright.dx.am/masukinnilai.php?r
    _username=" + user + "&r_idsoal=" + idsoal + "&r_ni
    lai=" + nilai1);
4.         yield return link.Send ();
5.         yield return new WaitForSeconds(delay);
6.         if (link.isError) {
7.             Debug.Log (link.error);
8.         }
9.     }

```

Kode Sumber 4.47 Implementasi *Insert* Nilai

```

1. <?php
2. header("Access-Control-Allow-Credentials: true");
3. header('Access-Control-Allow-Origin: *');
4. header('Access-Control-Allow-
  Methods: POST, GET, OPTIONS');
5. header('Access-Control-Allow-Headers: Accept, X-
  Access-Token, X-Application-Name, X-Request-Sent-
  Time');
6.
7. error_reporting(E_ALL ^ E_DEPRECATED);
8.
9. $Server = "fdb17.awardspace.net";
10. $User = "2326813_admin";
11. $Pass = "zzz1312000";
12. $Db = "2326813_admin";
13. $Connection = mysql_connect($Server, $User, $Pass,
  $Db);
14.
15. if(mysql_error()){
16.     die("Connection Failed!=". mysql_error());
17. }
18. mysql_select_db("2326813_admin", $Connection) or di
  e("Could not load to database" . mysql_error());
19.
20. $username = $_GET["r_username"];
21. $nilai = $_GET["r_nilai"];
22. $idsoal = $_GET["r_idsoal"];
23. $check = mysql_query("SELECT * FROM peserta WHERE `
  username` = '". $username. "'");
24. $numrows = mysql_num_rows($check);
25.
26. if($numrows == 0)
27. {
28.     die("Error");
29. }
30. else
31. {
32.     while($row = mysql_fetch_assoc($check))
33.     {
34.         if($username == $row['username'])

```

Kode Sumber 4.48 Kontrol *Insert Nilai* (Bagian Pertama)

```

35.         {
36.             $sql = mysql_query("UPDATE jawab INNER
JOIN peserta ON jawab.id=peserta.id SET jawab.skor
= ' ".$nilai." ' WHERE peserta.username = ' ".$username
e." ' and jawab.id_soal = ' ".$idsoal." ");
37.                                     die("berhasil");
38.         }
39.     else
40.     {
41.         die("error");
42.     }
43. }
44. }
45. ?>

```

Kode Sumber 4.49 Kontrol *Insert* Nilai (Bagian Kedua)

4.3.7 Implementasi Waktu Hitung Mundur Soal

Pada saat peserta mengerjakan soal, terdapat batas waktu untuk mengerjakan pada soal tersebut. Setiap soal mempunyai batas waktu yang berbeda-beda tergantung tingkat kesulitan soal.

```

1.  public float StartTimes, t;
2.  public string minutes, seconds;
3.
4.  StartTimes = 300;
5.
6.  t = StartTimes - Time.timeSinceLevelLoad;
7.  minutes = ((int)t / 60).ToString();
8.  seconds = (t % 60).ToString("f0");
9.
10. timerText.text = minutes + " : " + seconds;
11.
12. if (t <= 0)

```

Kode Sumber 4.50 Implementasi Waktu Hitung Mundur (Bagian Pertama)

```

13.      {
14.          timerText.text = "Waktu Habis";
15.          timerText.color = Color.red;
16.          PlayerPrefs.SetInt("scoreno1", -5);
17.
18.          int cekno1 = PlayerPrefs.GetInt ("score
19.          no1");
20.          int cekno2 = PlayerPrefs.GetInt ("score
21.          no2");
22.          int cekno0 = PlayerPrefs.GetInt ("score
23.          no0");
24.          int cekno3 = PlayerPrefs.GetInt ("score
25.          no3");
26.
27.          score1 = PlayerPrefs.GetInt("scoreno1")
28.          ;
29.          Debug.Log (PlayerPrefs.GetInt ("scoreno
30.          1"));
31.          user = PlayerPrefs.GetString("username"
32.          ).ToString();
33.          StartCoroutine(Postscore(user,idsoal, s
34.          core1, 5f));
35.
36.          string acak = scene.ToString();
37.          if (cekno1 != 0 && cekno2 != 0 && cekno
38.          0 != 0 && cekno3 != 0) {
39.              Application.LoadLevel ("pageSelesai
40.              ");
41.          }
42.          else if (cekno1 != 0 && cekno2 == 0 &&
43.          cekno0 == 0 && cekno3 == 0) {
44.              scene = number [UnityEngine.Random.
45.              Range (min, max)];
46.              acak = scene.ToString ();
47.              Debug.Log (acak);
48.              Application.LoadLevel (acak);
49.          } else if (cekno1 != 0 && cekno2 == 0 &
50.          & cekno0 != 0 && cekno3 == 0) {

```

Kode Sumber 4.51 Implementasi Waktu Hitung Mundur (Bagian Kedua)


```

38.         scene = number1 [UnityEngine.Random
        .Range (min1, max1)];
39.         print (scene);
40.         acak = scene.ToString ();
41.         Debug.Log (acak);
42.         Application.LoadLevel (acak);
43.     } else if (cekno1 != 0 && cekno2 != 0 &
    & cekno0 == 0 && cekno3 == 0) {
44.         scene = number2 [UnityEngine.Random
        .Range (min1, max1)];
45.         print (scene);
46.         acak = scene.ToString ();
47.         Debug.Log (acak);
48.         Application.LoadLevel (acak);
49.     } else if (cekno1 != 0 && cekno2 == 0 &
    & cekno0 == 0 && cekno3 != 0) {
50.         scene = number3 [UnityEngine.Random
        .Range (min1, max1)];
51.         print (scene);
52.         acak = scene.ToString ();
53.         Debug.Log (acak);
54.         Application.LoadLevel (acak);
55.     } else if (cekno0 == 0 && cekno2 != 0 &
    & cekno1 != 0 && cekno3 != 0) {
56.         Application.LoadLevel ("0");
57.     } else if (cekno0 != 0 && cekno2 == 0 &
    & cekno1 != 0 && cekno3 != 0) {
58.         Application.LoadLevel ("2");
59.     } else if (cekno0 != 0 && cekno2 != 0 &
    & cekno1 != 0 && cekno3 == 0) {
60.         Application.LoadLevel ("3");
61.     }

```

Kode Sumber 4.52 Implementasi Waktu Hitung Mundur (Bagian Ketiga)

Pada kode diatas, dijelaskan bahwa jika peserta tidak dapat menjawab hingga batas waktu yang diberikan pada masing-masing soal, maka jawaban peserta pada soal tersebut akan dianggap salah.

4.3.8 Implementasi Terjemahan Soal

Pada saat peserta mengerjakan soal, terdapat fitur untuk menerjemahkan soal. Karena *Google Translate API* sedang *deprecated* untuk Unity, maka penulis mendeteksi *string* dan menerjemahkan secara manual.

```

1.  public class OCL : MonoBehaviour
2.  {
3.      public delegate void ActiveChanged(bool isActive);
4.      public static ActiveChanged onActiveChanged
5.      ;
6.      public delegate void LanguageChanged(System
7.      Language oldLang, SystemLanguage newLang);
8.      public static LanguageChanged onLanguageCha
9.      nged;
10.     public delegate void LanguagesChanged();
11.     public static LanguagesChanged onLanguagesC
12.     hanged;
13.     public delegate void LocalizationChanged(ob
14.     ject id, SystemLanguage language, object newValue);
15.     public static LocalizationChanged onLocaliz
16.     ationChanged;
17.     protected static LocalizationSetup runtimeS
18.     etup;
19.     protected static string parameterPattern =
20.     @"\${{([0-9])*}}";
21.     protected static List<string> stringsWithPa
22.     rameters;
23.     protected static List<Regex> stringsWithPar
24.     ametersRegex;

```

Kode Sumber 4.53 Implementasi Terjemahkan Soal (Bagian Pertama)

```

17.
18.         static OCL()
19.         {
20.             LocalizationSetup persistedSetup = Resources.Load<LocalizationSetup>(LocalizationSetup.setupResourceName);
21.             if (persistedSetup == null)
22.             {
23.                 Debug.LogWarning("[OCL] No setup as set <" + LocalizationSetup.setupResourceName + "> found on disk. Initializing deactivated empty setup.");
24.
25.                 runtimeSetup = ScriptableObject.CreateInstance<LocalizationSetup>();
26.                 runtimeSetup.active = false;
27.                 runtimeSetup.AddLanguage(runtimeSetup.defaultLanguage);
28.             }
29.             else
30.             {
31.                 runtimeSetup = (LocalizationSetup) persistedSetup.Clone();
32.                 runtimeSetup.LoadSubAssets(true);
33.             }
34.             initStringsWithParameters();
35.         }
36.
37.         protected static void initStringsWithParameters()
38.         {
39.             stringsWithParameters = new List<string>();
40.             stringsWithParametersRegex = new List<Regex>();
41.
42.             foreach(string stringId in runtimeSetup.GetIds<string>())
43.             {

```

Kode Sumber 4.54 Implementasi Terjemahkan Soal (Bagian Kedua)

```

44.         string[] substrings = Regex.Split(s
tringId, parameterPattern);
45.         if (substrings.Length > 1)
46.         {
47.
48.             int index = 0;
49.             string res = "";
50.             foreach (string match in substr
ings)
51.             {
52.                 if (index % 2 == 0)
53.                 {
54.                     res += match;
55.                 }
56.                 else
57.                 {
58.                     res += "(.*)";
59.                 }
60.
61.                 index++;
62.             }
63.             Regex stringRegex = new Regex(r
es, RegexOptions.IgnoreCase);
64.             stringsWithParameters.Add(strin
gId);
65.             stringsWithParametersRegex.Add(
stringRegex);
66.         }
67.     }
68. }
69.
70. public static bool IsActive()
71. {
72.     return runtimeSetup.active;
73. }
74. /// <param name="value"></param>
75. public static void SetActive(bool value)
76. {
77.     runtimeSetup.active = value;

```

Kode Sumber 4.55 Implementasi Terjemahkan Soal (Bagian Ketiga)

```
78.         if(onActiveChanged != null)
79.         {
80.             onActiveChanged(IsActive());
81.         }
82.     }
83. public static SystemLanguage GetLanguage()
84.     {
85.         if (IsLanguageAuto())
86.         {
87.             return Application.systemLanguage;
88.         }
89.         else
90.         {
91.             return runtimeSetup.forcedLanguage;
92.         }
93.     }
```

Kode Sumber 4.56 Implementasi Terjemahkan Soal (Bagian Keempat)

BAB V

PENGUJIAN DAN EVALUASI

Pada bab ini membahas mengenai uji coba dan evaluasi aplikasi. Uji coba dilakukan menggunakan metode *black box* (kotak hitam), yaitu pengujian yang berfokus pada spesifikasi fungsional dari perangkat lunak berdasarkan skenario yang telah ditentukan.

5.1 Lingkungan Uji Coba

Pada proses pengujian perangkat lunak, dibutuhkan suatu lingkungan pengujian yang sesuai dengan standar kebutuhan. Lingkungan pengujian dalam tugas akhir ini dilakukan pada setiap kasus penggunaan. Spesifikasi masing-masing lingkungan pengujian dijabarkan pada Tabel 5.1.

Tabel 5.1 Lingkungan Pengujian Fungsionalitas Perangkat Lunak

Spesifikasi	Deskripsi
Jenis Perangkat	Laptop
Merek Perangkat	MSI
Sistem Operasi	Sistem operasi Windows 10 (64-bit)
CPU	Prosesor Intel(R) Core(TM) i7-6700HQ CPU @ 2.60GHz
RAM	12 GB

5.2 Pengujian Fungsionalitas

Pengujian fungsionalitas aplikasi ini dapat dilakukan secara mandiri. Pengujian ini bertujuan untuk mengetahui kesesuaian keluaran dari tiap tahap dan langkah penggunaan fitur terhadap skenario yang dipersiapkan. Skenario yang dibuat mengacu pada kasus penggunaan yang telah dijelaskan pada subbab 3.1.3.4.

Skenario uji coba fungsionalitas yang dilakukan terhadap aplikasi yang dibangun dijelaskan pada Tabel 5.2.

Tabel 5.2 Skenario Uji Coba Fungsionalitas

Kode Uji Coba	Nama Uji Coba
UC-0001	Uji melihat tentang bebras
UC-0002	Uji menambah detail peserta
UC-0003	Uji mengerjakan soal
UC-0004	Uji melihat nilai
UC-0005	Uji melihat peringkat

Setiap skenario akan dijelaskan mengenai kondisi awal, masukkan, dan keluaran yang diharapkan, kondisi akhir, dan hasil uji coba. Berikut ini merupakan penjabaran hasil setiap uji coba yang dilakukan.

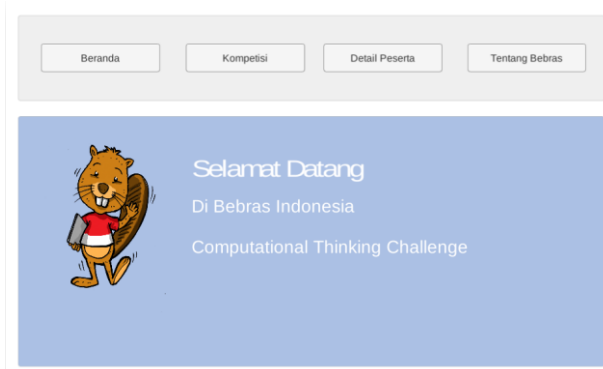
5.2.1 Pengujian Fungsionalitas Melihat Tentang Bebras

Tabel 5.3 Uji Coba Melihat Tentang Bebras

No.Pengujian	UJ-UC-0001
Skenario Pengujian	Aplikasi menampilkan informasi tentang bebras ketika masuk ke halaman Tentang Bebras
Kondisi Awal	Aplikasi menampilkan halaman Beranda yang terdapat beberapa menu, yaitu “Beranda”, “Kompetisi”, “Detail Peserta”, “Tentang Bebras”
Aksi yang dilakukan	Memilih tombol “Tentang Bebras”
Hasil yang diharapkan	Aplikasi dapat menampilkan informasi tentang bebras
Hasil yang diperoleh	Aplikasi menampilkan informasi tentang bebras
Hasil Pengujian	Berhasil

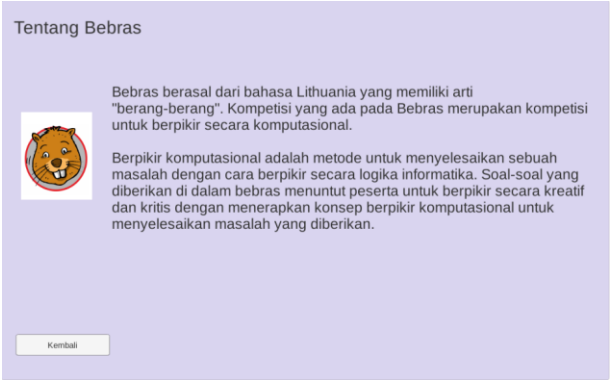
Langkah-langkah UJ-UC-0001:

1. Peserta masuk ke halaman Beranda seperti pada Gambar 5.1.



Gambar 5.1 Halaman Beranda

2. Peserta menekan tombol yang bertuliskan “Tentang Bebras”.
3. Sistem menampilkan informasi tentang bebras seperti pada Gambar 5.2 yang menunjukkan bahwa peserta telah sukses melihat informasi tentang bebras.



Gambar 5.2 Halaman Tentang Bebras

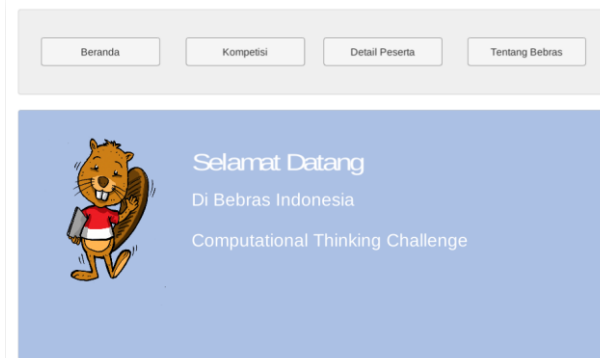
5.2.2 Pengujian Fungsionalitas Melihat Detail Peserta

Tabel 5.4 Uji Coba Melihat Detail Peserta

No.Pengujian	UJ-UC-0002
Skenario Pengujian	Aplikasi menampilkan data peserta ketika masuk ke halaman Detail Peserta
Kondisi Awal	Aplikasi menampilkan halaman Beranda yang terdapat beberapa menu, yaitu “Beranda”, “Kompetisi”, “Detail Peserta”, “Tentang Bebras”
Aksi yang dilakukan	Memilih tombol “Detail Peserta”
Hasil yang diharapkan	Aplikasi dapat menampilkan data peserta
Hasil yang diperoleh	Aplikasi menampilkan data peserta
Hasil Pengujian	Berhasil

Langkah-langkah UJ-UC-0002:

1. Peserta masuk ke halaman Beranda.



Gambar 5.3 Halaman Beranda

2. Peserta menekan tombol yang bertuliskan “Detail Peserta”.
3. Sistem menampilkan data peserta yang menunjukkan bahwa peserta telah sukses melihat data peserta.



Gambar 5.4 Halaman Detail Peserta

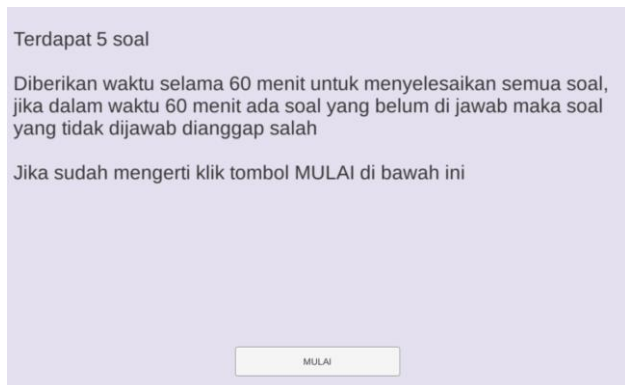
5.2.3 Pengujian Fungsionalitas Mengerjakan Soal

Tabel 5.5 Uji Coba Mengerjakan Soal

No.Pengujian	UJ-UC-0003
Skenario Pengujian	Peserta dapat mengerjakan soal pada sistem
Kondisi Awal	Aplikasi menampilkan halaman Penjelasan Soal yang terdapat menu “Mulai”
Aksi yang dilakukan	Memilih tombol “Mulai”
Hasil yang diharapkan	Peserta dapat mengerjakan soal pada sistem
Hasil yang diperoleh	Peserta mengerjakan soal pada sistem
Hasil Pengujian	Berhasil

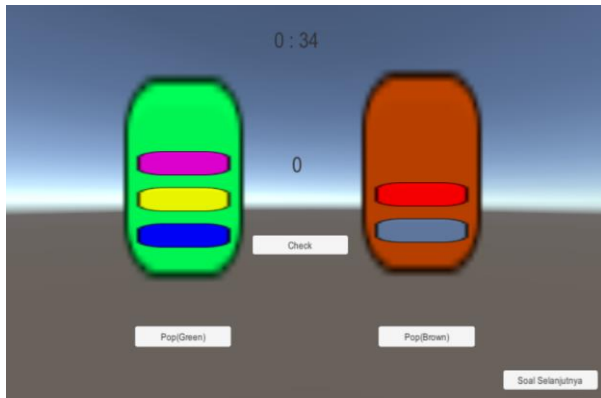
Langkah-langkah UJ-UC-0003:

1. Peserta masuk ke halaman Penjelasan Soal.



Gambar 5.5 Halaman Penjelasan Soal

2. Peserta menekan tombol yang bertuliskan “Mulai”.
3. Sistem menampilkan soal dan pilihan jawaban yang menunjukkan bahwa peserta telah sukses melihat soal.



Gambar 5.6 Halaman Soal

4. Peserta dapat mengerjakan soal sesuai dengan petunjuk yang diberikan.
5. Peserta yang sudah selesai mengerjakan soal dapat menyimpan nilai dan melanjutkan ke soal berikutnya dengan memilih tombol “Soal Selanjutnya”.

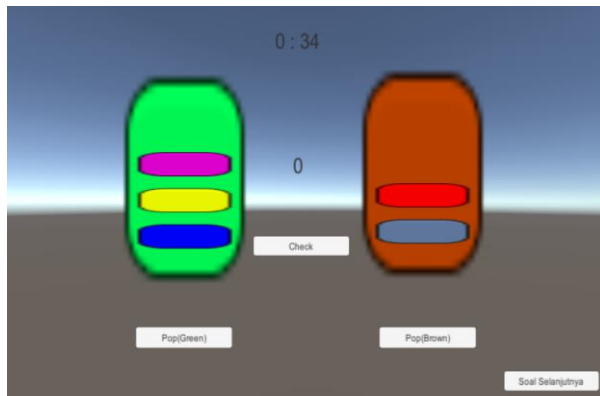
5.2.4 Pengujian Fungsionalitas Melihat Nilai

Tabel 5.6 Uji Coba Melihat Nilai

No.Pengujian	UJ-UC-0004
Skenario Pengujian	Aplikasi menampilkan data nilai ketika masuk ke halaman Selesai
Kondisi Awal	Aplikasi menampilkan halaman Soal yang terdapat menu yaitu “Selanjutnya”
Aksi yang dilakukan	Memilih tombol “Selanjutnya”
Hasil yang diharapkan	Aplikasi dapat menampilkan data nilai yang didapat oleh peserta
Hasil yang diperoleh	Aplikasi menampilkan data nilai yang didapat oleh peserta
Hasil Pengujian	Berhasil

Langkah-langkah UJ-UC-0004:

1. Peserta masuk ke halaman soal.



Gambar 5.7 Halaman Soal

2. Peserta menekan tombol yang bertuliskan “Soal Selanjutnya”.
3. Jika semua soal telah dikerjakan, maka sistem akan menampilkan data nilai yang didapat oleh peserta yang menunjukkan bahwa peserta telah sukses melihat data nilai.



Gambar 5.8 Halaman Selesai

5.2.5 Pengujian Fungsionalitas Halaman *Ranking*

Tabel 5.7 Uji Coba Melihat *Ranking*

No.Pengujian	UJ-UC-0005
Skenario Pengujian	Aplikasi menampilkan data peringkat ketika masuk ke halaman Ranking
Kondisi Awal	Aplikasi menampilkan halaman Selesai yang terdapat beberapa menu, yaitu “Ranking” dan “Log Out”
Aksi yang dilakukan	Memilih tombol “Ranking”
Hasil yang diharapkan	Aplikasi dapat menampilkan data peringkat berupa data semua nilai peserta
Hasil yang diperoleh	Aplikasi menampilkan data peringkat berupa data semua nilai peserta
Hasil Pengujian	Berhasil

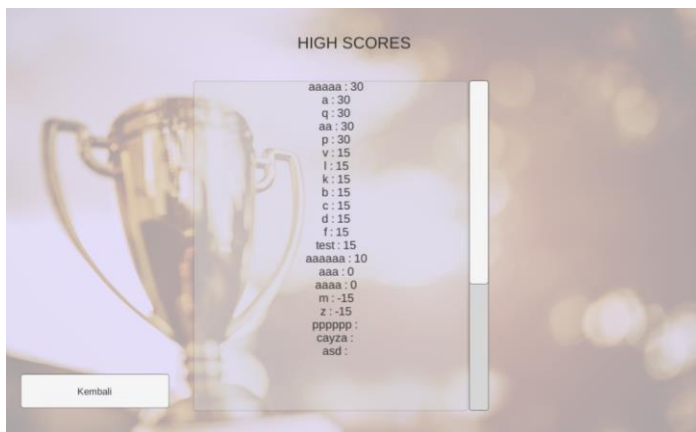
Langkah-langkah UJ-UC-0005:

1. Peserta masuk ke halaman selesai.



Gambar 5.9 Halaman Selesai

2. Peserta menekan tombol yang bertuliskan “RANKING”.
3. Sistem menampilkan data peringkat berupa data semua nilai peserta yang menunjukkan bahwa peserta telah sukses melihat peringkat.



Gambar 5.10 Halaman *Ranking*

5.3 Evaluasi Pengujian

Pada subbab ini akan diberikan hasil evaluasi dari pengujian-pengujian yang telah dilakukan. Evaluasi yang diberikan meliputi evaluasi pengujian fungsional, serta evaluasi pengujian aplikasi terhadap pengguna.

5.3.1 Evaluasi Pengujian Fungsionalitas

Evaluasi pengujian fungsionalitas dilakukan dengan menampilkan data rekapitulasi aplikasi yang telah dipaparkan. Dalam hal ini, rekapitulasi disusun dalam bentuk tabel yang dapat dilihat pada. Dari data yang terdapat pada tabel tersebut, diketahui bahwa aplikasi yang dibuat telah memenuhi kasus penggunaan yang telah ditentukan.

Tabel 5.8 Rekapitulasi Hasil Uji Fungsionalitas

Kode	Uji Coba	Hasil
UJ-UC-0001	Uji melihat tentang bebras	Berhasil
UJ-UC-0002	Uji menambah detail peserta	Berhasil
UJ-UC-0003	Uji mengerjakan soal	Berhasil
UJ-UC-0004	Uji melihat nilai	Berhasil
UJ-UC-0005	Uji melihat peringkat	Berhasil

5.3.2 Evaluasi Pengujian Aplikasi Terhadap Pengguna

Aplikasi ini telah diujikan kepada pengguna. Setelah mencoba aplikasi, pengguna diberikan kuesioner yang diisi berdasarkan pendapat mereka.. Hasil dari kuesioner dapat dilihat pada Tabel 5.9.

Tabel 5.9 Hasil Kuesioner Pengguna

No	Pertanyaan	Jumlah				
		STS	TS	C	S	SS
1	Aplikasi memiliki tampilan dan desain yang menarik	0	0	0	3	0
2	Aplikasi memiliki menu yang mudah digunakan	0	0	0	2	1
3	Aplikasi memiliki tata letak tombol yang sesuai	0	0	1	2	0
4	Aplikasi nyaman untuk digunakan	0	0	0	3	0
5	Aplikasi diakses dengan cepat	0	0	1	0	2
6	Aplikasi memberikan soal yang menarik dan menantang	0	0	0	0	3
7	Aplikasi memberikan soal yang sesuai dengan materi	0	0	0	2	1

Tabel 5.10 Rata-Rata Nilai Hasil Kuesioner Pengguna

No	Parameter Antarmuka	N
1	Aplikasi memiliki tampilan dan desain yang menarik	4
2	Aplikasi memiliki menu yang mudah digunakan	4,3
3	Aplikasi Memiliki tata letak tombol yang sesuai	3,6
	Parameter Performa dan Kenyamanan	
4	Aplikasi nyaman untuk digunakan	4
5	Aplikasi diakses dengan cepat	4,3
No	Parameter Materi Soal	N
5	Aplikasi memberikan Soal yang Menarik	4,6

Berdasarkan Tabel 5.10, pada parameter antarmuka poin nomor 1 dan 2, aplikasi memiliki tampilan yang menarik dan memiliki menu yang mudah digunakan. Sedangkan untuk poin nomor 3, tata letak tombol pada aplikasi cukup sesuai dengan mendapatkan rata-rata nilai 3,6 berdasarkan kuesioner pengguna. Pada parameter performa dan kenyamanan, dapat disimpulkan bahwa aplikasi nyaman untuk digunakan dan pengguna dapat mengakses aplikasi dengan cepat. Untuk parameter materi soal, dapat disimpulkan bahwa pengguna merasa tertantang dan tertarik untuk menyelesaikan soal yang diberikan oleh aplikasi.

BAB VI

KESIMPULAN DAN SARAN

Bab ini membahas mengenai kesimpulan yang didapatkan dari uji tujuan pembuatan perangkat lunak dan hasil uji coba yang telah dilakukan. Selain kesimpulan, terdapat pula saran-saran untuk pengembangan perangkat lunak lebih lanjut.

6.1. Kesimpulan

Dari proses pengerjaan tugas akhir selama tahap analisis, desain, implementasi, hingga pengujian didapatkan kesimpulan sebagai berikut:

1. Berdasarkan hasil uji coba fungsionalitas, aplikasi berhasil dibangun sesuai rancangan.
2. Berdasarkan kuesioner, responden cukup setuju antarmuka aplikasi nyaman, dengan nilai 3,9 dari nilai maksimal 5.
3. Berdasarkan kuesioner, responden setuju performa aplikasi baik, dengan nilai 4,1 dari nilai maksimal 5.
4. Berdasarkan kuesioner, responden setuju aplikasi memberi soal yang menantang, dengan persentase 4,6 dari nilai maksimal 5.

6.2. Saran

Berikut merupakan beberapa saran untuk perbaikan dan pengembangan aplikasi dimasa yang akan datang:

1. Gambar-gambar yang dipakai pada soal dibuat semenarik mungkin agar pengguna tertarik.
2. Menggunakan jasa *web hosting* yang berbayar agar mengakses basis data lebih cepat.

DAFTAR PUSTAKA

- [1] “Beranda | Bebras Indonesia.” [Daring]. Tersedia pada: <http://bebras.or.id/>. [Diakses: 02-Jun-2017].
- [2] “What is Bebras | www.bebas.org.” [Daring]. Tersedia pada: <http://bebras.org/>. [Diakses: 02-Jun-2017].
- [3] “TOKI SELENGGARAKAN BEBRAS INDONESIA CHALLENGE 2016 » Tim Olimpiade Komputer Indonesia.” [Daring]. Tersedia pada: <http://www.toki.or.id/2016/toki-selenggarakan-bebas-indonesia-challenge-2016/>. [Diakses: 02-Jun-2017].
- [4] “Mengenal WebGL - CodePolitan.com.” [Daring]. Tersedia pada: <https://www.codepolitan.com/mengenal-webgl>. [Diakses: 02-Jun-2017].
- [5] “Combinatorics,” *Wikipedia*. 24-Nov-2016.
- [6] “Unity (game engine),” *Wikipedia*. 28-Jun-2017.
- [7] “C sharp,” *Wikipedia bahasa Indonesia, ensiklopedia bebas*. 13-Jun-2013.
- [8] L. Webster, “Visual Studio IDE, Code Editor, Team Services, & Mobile Center,” *Visual Studio*, 22-Jun-2017. [Daring]. Tersedia pada: <https://www.visualstudio.com/>. [Diakses: 11-Jul-2017].
- [9] “Lightbot.” [Daring]. Tersedia pada: <https://lightbot.com/>. [Diakses: 17-Jul-2017].

LAMPIRAN A

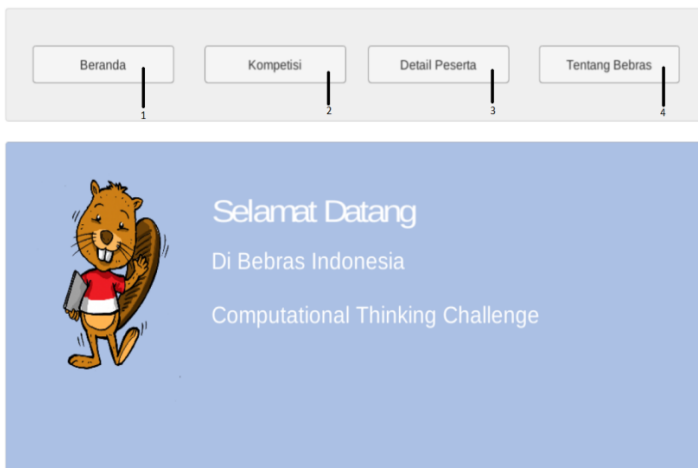
PANDUAN PENGGUNAAN APLIKASI

Berikut ini adalah lampiran panduan penggunaan aplikasi mulai dari pendaftaran hingga selesai mengerjakan soal.

- **Mengakses Aplikasi**

Berikut ini akan dijelaskan cara pengguna mengakses aplikasi:

1. Buka aplikasi pada perangkat Android atau buka pada *browser*.
2. Pada halaman awal dari aplikasi terdapat empat menu yaitu Beranda, Kompetisi, Detail Peserta, dan Tentang Bebras.



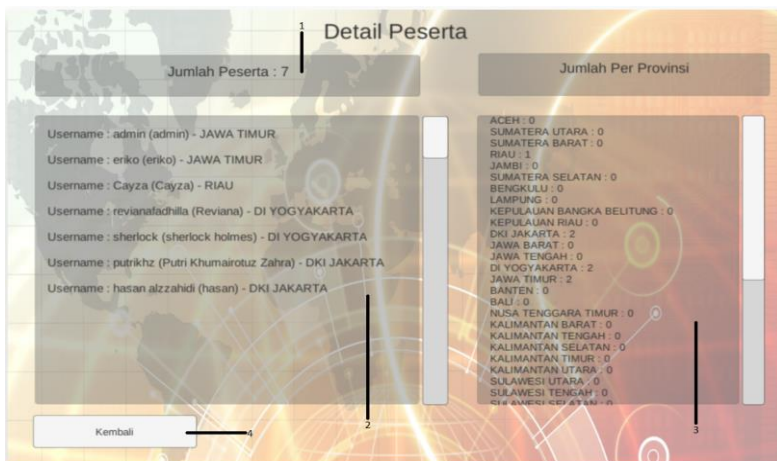
Gambar A.1 Halaman Beranda

1. Tombol menuju halaman Beranda.
2. Tombol menuju halaman Kompetisi.
3. Tombol menuju halaman Detail Peserta.
4. Tombol menuju halaman Tentang Bebras

- Mengakses Halaman Detail Peserta

Berikut ini akan dijelaskan cara pengguna mengakses halaman kompetisi dan dijelaskan fitur pada halaman detail peserta:

1. Jika pada halaman beranda pengguna memilih Tombol Detail Peserta maka pengguna akan diarahkan menuju halaman Detail Peserta.
2. Pada halaman Detail Peserta terdapat informasi seperti jumlah peserta, detail peserta yang berisi *username*, nama dan asal provinsi peserta.



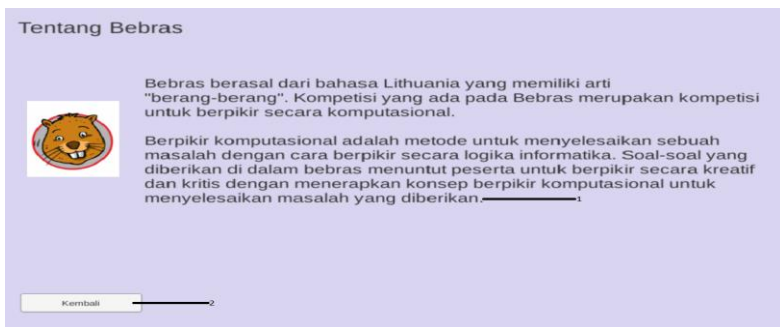
Gambar A.2 Halaman Detail Peserta

1. Teks untuk menampilkan jumlah peserta.
2. Teks untuk menampilkan detail peserta.
3. Teks untuk menampilkan jumlah peserta per provinsi.
4. Tombol kembali ke halaman beranda.

- Mengakses Halaman Tentang Bebras

Berikut ini akan dijelaskan cara pengguna mengakses halaman tentang bebras:

1. Jika pada halaman beranda pengguna memilih Tombol Tentang Bebras maka pengguna akan diarahkan menuju Halaman Tentang Bebras.
2. Pada Halaman Tentang Bebras hanya berisi tulisan dan tombol kembali ke Halaman Beranda.



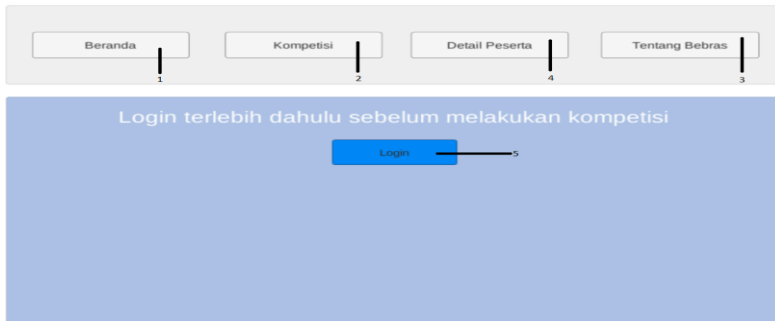
Gambar A.3 Halaman Tentang Bebras

1. Teks Tentang Bebras
2. Tombol kembali ke halaman beranda

- Mengakses Halaman Kompetisi

Berikut ini akan dijelaskan cara pengguna mengakses Halaman Kompetisi:

1. Jika pada halaman beranda pengguna memilih Tombol Kompetisi maka pengguna akan diarahkan menuju Halaman Kompetisi.
2. Pada Halaman Kompetisi terdapat sebuah tombol untuk masuk halaman *login*.



Gambar A.4 Halaman Kompetisi

1. Tombol untuk menuju Halaman Beranda.
2. Tombol untuk menuju Halaman Kompetisi.
3. Tombol untuk menuju Halaman Tentang Bebras.
4. Tombol untuk menuju Halaman Detail Peserta.
5. Tombol untuk menuju Halaman *Login*.

- **Mengakses Halaman *Login***

Berikut ini akan dijelaskan cara pengguna mengakses Halaman *Login* dan dijelaskan fitur pada Halaman *Login*:

1. Jika pada Halaman Kompetisi pengguna memilih tombol *login* maka pengguna akan diarahkan menuju halaman *login*.
2. Pada halaman *login* terdapat tombol *register* jika pengguna belum memiliki akun untuk *login*. Dan terdapat dua *input fields* untuk memasukkan *username* dan *password*.



Gambar A.5 Halaman *Login*

- | |
|--|
| <ol style="list-style-type: none">1. <i>Input Field</i> untuk memasukkan <i>username</i>,2. <i>Input Field</i> untuk memasukkan <i>password</i>.3. Tombol untuk melakukan <i>login</i>.4. Tombol untuk melakukan <i>register</i>.5. Tombol untuk menuju Halaman Beranda. |
|--|

- Mengakses Halaman Register

Berikut ini akan dijelaskan cara pengguna mengakses halaman *register* dan dijelaskan fitur pada halaman *register*:

1. Saat pengguna berada pada halaman *login* dan belum memiliki akun, maka pengguna diminta untuk membuat akun terlebih dahulu dengan cara memilih tombol *register* pada halaman *login*.
2. Saat pengguna berada pada halaman *register*, pengguna diharuskan mengisi beberapa *field* untuk data peserta.

The image shows a web form titled "REGISTER" on a light gray background. The form contains several input fields and two buttons. Numbered callouts point to specific elements: 1 points to the "Username" field, 2 to the "Nama" field, 3 to the "Password" field, 4 to the "Ulangi Password" field, 5 to the "Pilih Provinsi Anda" dropdown menu, 6 to the green "Register" button, and 7 to the "Kembali" button.

Callout Number	Field/Element
1	Username
2	Nama
3	Password
4	Ulangi Password
5	Pilih Provinsi Anda
6	Register
7	Kembali

Gambar A.6 Halaman *Register*

1. *Input Field* untuk *Username*.
2. *Input Field* untuk Nama.
3. *Input Field* untuk *Password*.
4. *Input Field* untuk Ulangi *Password*.
5. *Input Field* untuk memilih provinsi asal.
6. Tombol untuk *register* / daftar.
7. Tombol untuk kembali ke halaman *login*.

- Mengakses Halaman Penjelasan Soal

Berikut ini akan dijelaskan cara pengguna mengakses halaman penjelasan soal:

1. Setelah *login* maka pengguna akan diarahkan pada halaman penjelasan soal.
2. Halaman penjelasan soal berisi teks penjelasan dan tombol untuk lanjut mengerjakan soal.



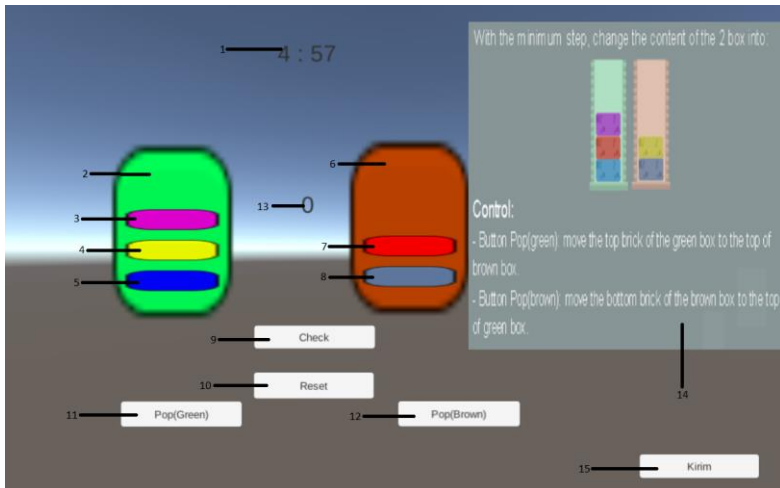
Gambar A.7 Halaman Penjelasan Soal

1. Teks Penjelasan Soal.
2. Tombol untuk mulai mengerjakan soal.

- Mengakses Halaman Soal

Berikut ini akan dijelaskan cara pengguna mengakses halaman soal beserta beberapa tombol yang ada pada halaman soal:

1. Setelah peserta mengakses halaman penjelasan soal lalu pengguna memilih tombol Mulai. Setelah itu akan muncul soal yang harus dikerjakan oleh pengguna.
2. Penulis memilih salah satu dari empat soal sebagai contoh pada lampiran ini.



Gambar A.8 Halaman Soal

1. Teks untuk waktu mengerjakan soal.
2. *Game Object* untuk wadah berisi piringan-piringan yang mengimplementasi operasi *stack*.
3. Piringan berwarna ungu.
4. Piringan berwarna kuning.
5. Piringan berwarna biru.
6. *Game Object* untuk wadah berisi piringan-piringan yang mengimplementasi operasi *queue*.
7. Piringan berwarna merah.
8. Piringan berwarna abu-abu.
9. Tombol *Check* jawaban.
10. Tombol *reset* jawaban.
11. Tombol untuk operasi *pop* pada wadah berwarna biru.
12. Tombol untuk operasi *queue* pada wadah berwarna coklat.
13. Teks untuk menampilkan jumlah langkah pengguna.
14. Gambar untuk penjelasan soal.
15. Tombol untuk selesai mengerjakan soal. Bisa menuju halaman selesai atau soal selanjutnya.

- Mengakses Halaman Selesai

Berikut ini akan dijelaskan cara pengguna mengakses halaman selesai:

1. Setelah peserta mengerjakan soal sejumlah empat soal, maka peserta akan diarahkan menuju halaman selesai.

2. Halaman selesai menampilkan nilai total yang didapat oleh peserta, tombol *ranking* dan tombol *log out*.



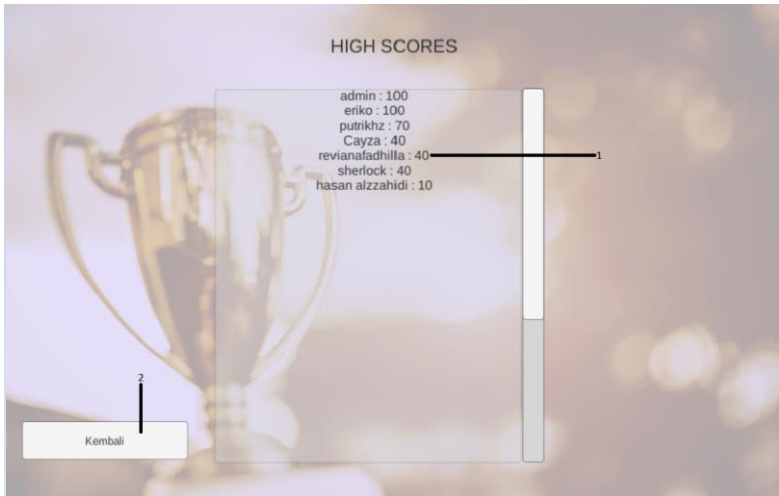
Gambar A.9 Halaman Selesai

1. Teks untuk menampilkan total nilai yang didapat oleh peserta.
2. Tombol untuk menuju halaman *ranking*.
3. Tombol untuk *log out* dan menuju halaman beranda.

- Mengakses Halaman *Ranking*

Berikut ini akan dijelaskan cara pengguna mengakses halaman *ranking*:

1. Halaman *ranking* akan muncul ketika peserta memilih tombol *ranking* pada halaman selesai.
2. Halaman *ranking* berisi *username* dan total nilai yang terurut dari tertinggi hingga terendah.



Gambar A.10 Halaman *Ranking*

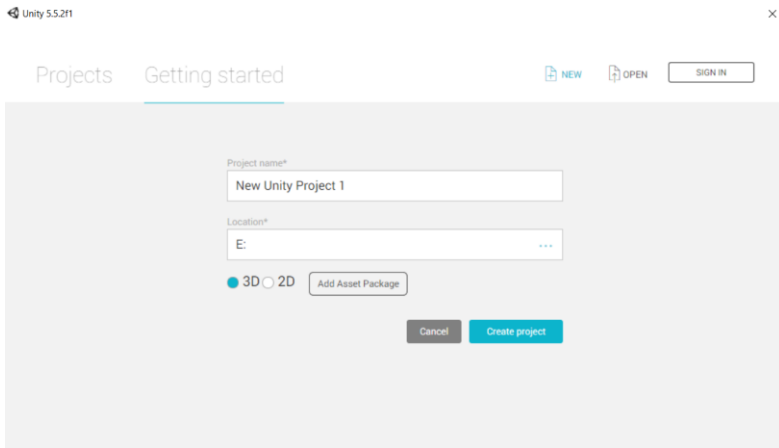
1. Teks untuk menampilkan *ranking* peserta dari nilai tertinggi ke terendah.
2. Tombol untuk kembali ke halaman selesai.

LAMPIRAN B

PANDUAN PEMBUATAN SOAL

Berikut ini adalah lampiran untuk urutan cara pembuatan soal pada Unity. Pada bab ini, penulis lampirkan cara membuat satu soal dari dua soal yang dibuat oleh penulis.

- Membuat Project di Unity

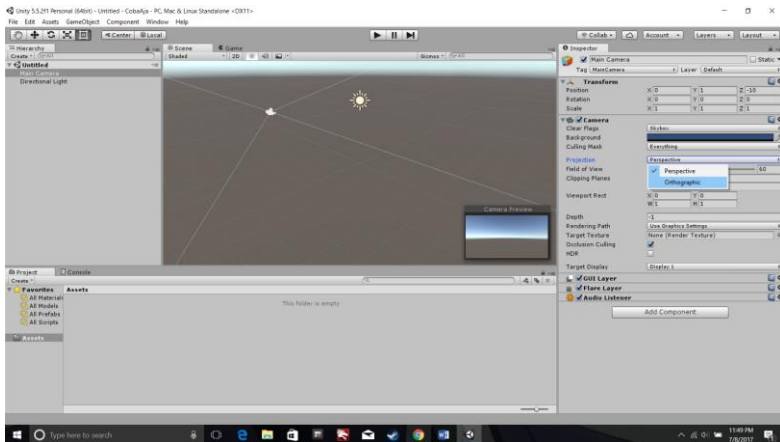


Gambar B.1 Tampilan Awal Unity

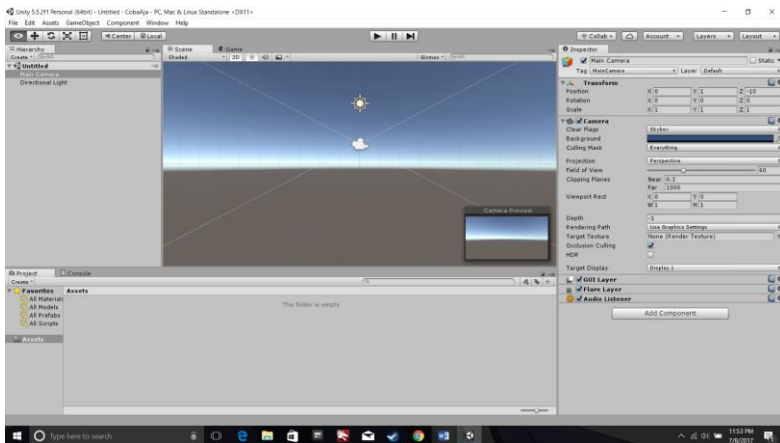
1. Buat *project* baru pada Unity dengan cara membuka aplikasi unity, lalu klik tulisan “NEW” maka akan muncul jendela seperti gambar diatas.
2. Terdapat dua *fields*, yaitu nama *project* dan lokasi penyimpanan *project*. Pembuat soal dapat memilih dengan bebas nama *project* dan lokasi penyimpanan.
3. Terdapat pilihan “3D” atau “2D”. Jika pembuat soal memilih “3D” maka *project* akan dibuat dalam bentuk 3

dimensi. Penulis menyarankan untuk membuat project dalam 3 dimensi karena pada *project* 3 dimensi dapat diubah menjadi 2 dimensi.

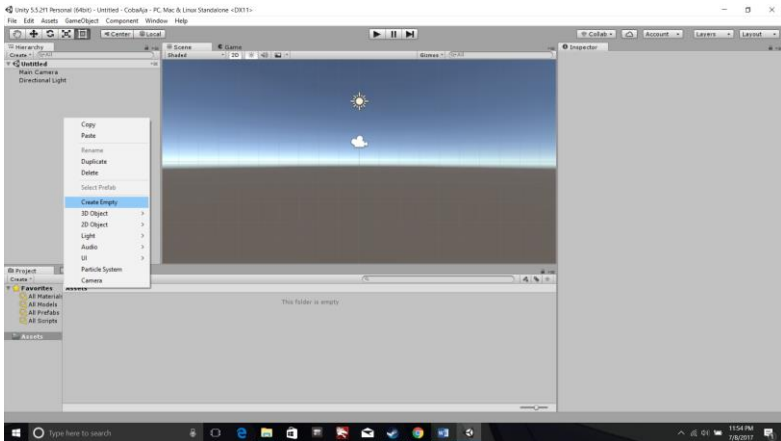
- Membuat *Game Object* pada unity



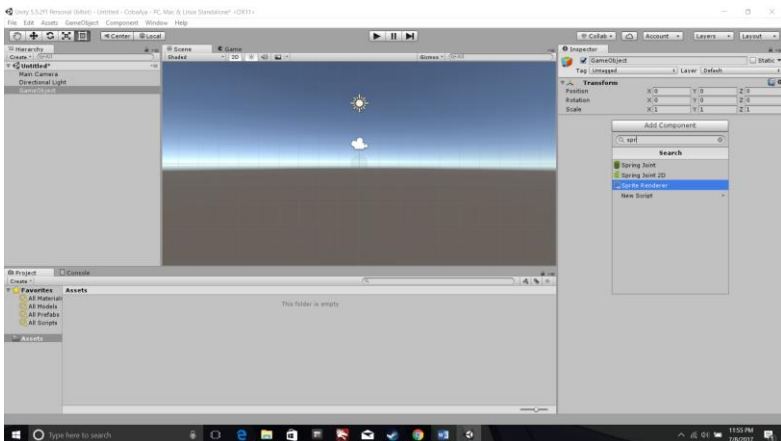
Gambar B.2 Tampilan Awal



Gambar B.3 Ubah *Perspective* Kamera



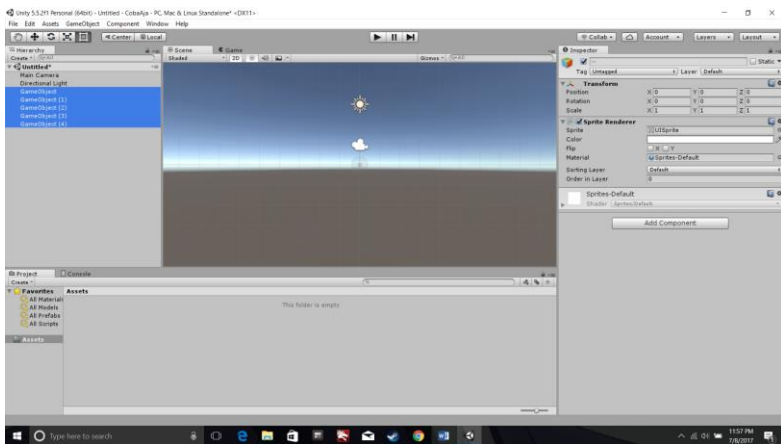
Gambar B.4 Membuat *Game Object*



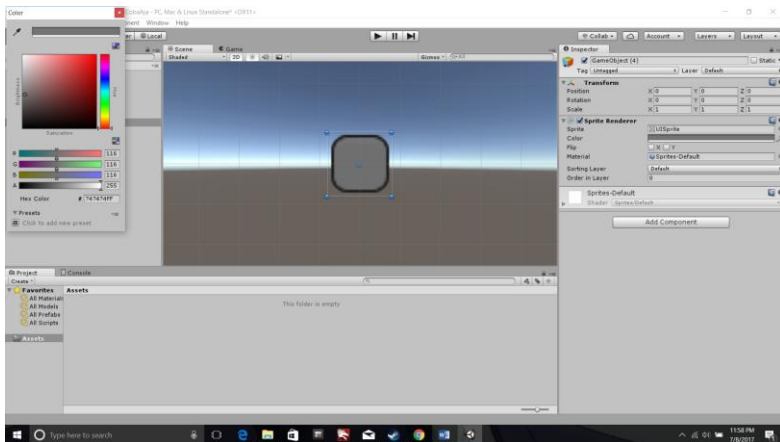
Gambar B.5 Menambah *Sprite Renderer* untuk *Game Object*



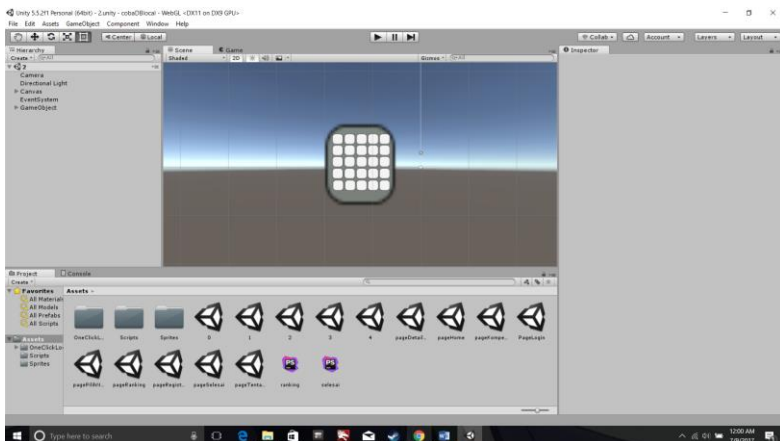
Gambar B.6 Memilih *UISprite* untuk *Sprite Renderer*



Gambar B.7 Menduplikasi *Game Object*



Gambar B.8 Mewarnai Game Object

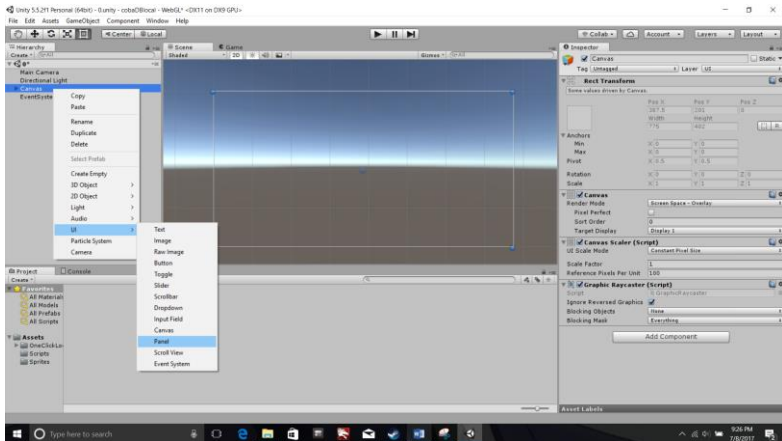


Gambar B.9 Mengatur Letak Game Object

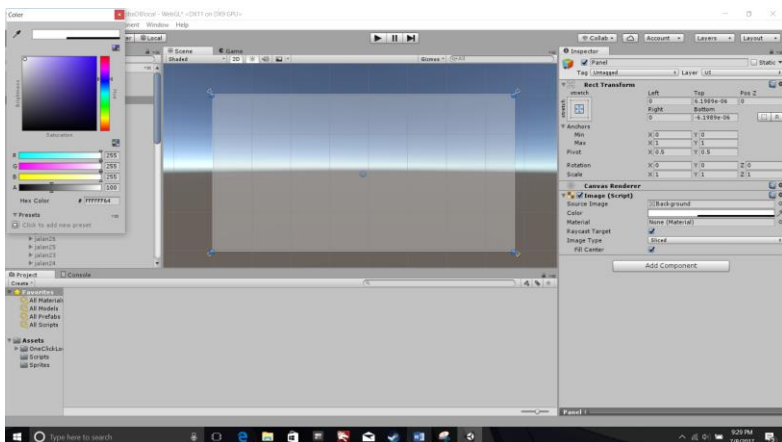
1. Pilih *Main Camera* pada *hierarchy*, lalu pada *inspector* yang terletak di sebelah kanan, ubah *projection* dari *Perspective* menjadi *Orthographic*.
2. Ubah *perspective* kamera menjadi 2 dimensi seperti pada Gambar B.3.

3. Klik kanan pada kolom *hierarchy*, lalu pilih *Create Empty* untuk membuat *Game Object*. Setelah itu, *Game Object* akan muncul pada kolom *hierarchy* seperti pada Gambar B.4.
4. Setelah muncul *Game Object* pada *hierarchy*, klik *Game Object* pada *hierarchy* lalu pada *inspector* di kolom sebelah kanan, pilih *Add Component* lalu cari *Sprite Renderer* seperti pada Gambar B.5.
5. Setelah menambahkan *Sprite Renderer* pada *Game Object*, terdapat baris bertuliskan “Sprite” dalam komponen *Sprite Renderer*. Klik bulatan kecil di sebelah baris bertuliskan “Sprite” lalu pilih “UISprite” seperti pada Gambar B.6.
6. Lalu duplikasi sebanyak yang diinginkan pembuat soal. Penulis membuat 26 *Game Object* dengan rincian 25 *Game Object* diibaratkan lampu dan 1 *Game Object* dijadikan *background* seperti pada Gambar B.7.
7. Pembuat soal dapat memberi warna pada *Game Object*. Cara memberi warna ditunjukkan pada Gambar B.8.
8. Setelah semua *Game Object* selesai diduplikasi dan diberi warna, maka atur letak *Game Object* sedemikian rupa sehingga seperti pada Gambar B.9.

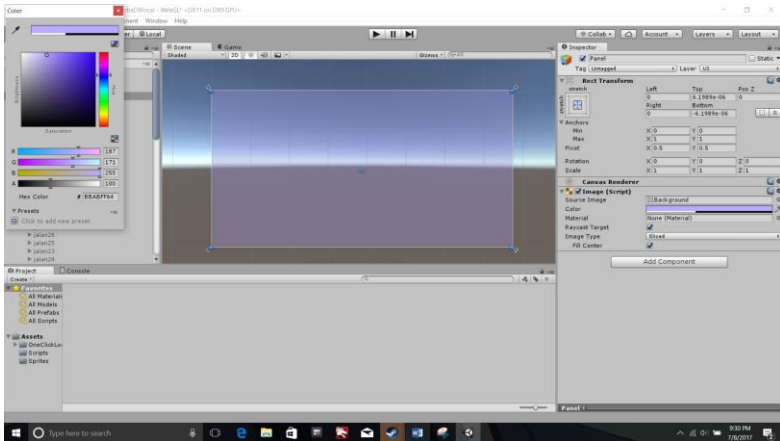
- Membuat Penjelasan Soal beserta Tombol yang Diperlukan



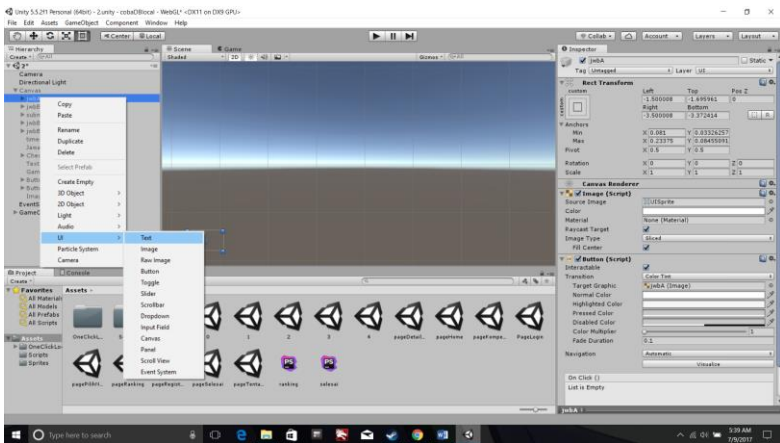
Gambar B.10 Membuat Canvas



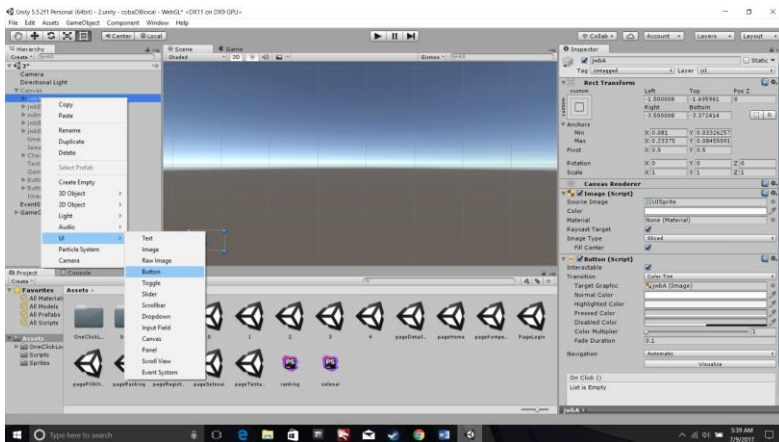
Gambar B.11 Memilih Warna Canvas



Gambar B.12 Konfirmasi Warna Canvas



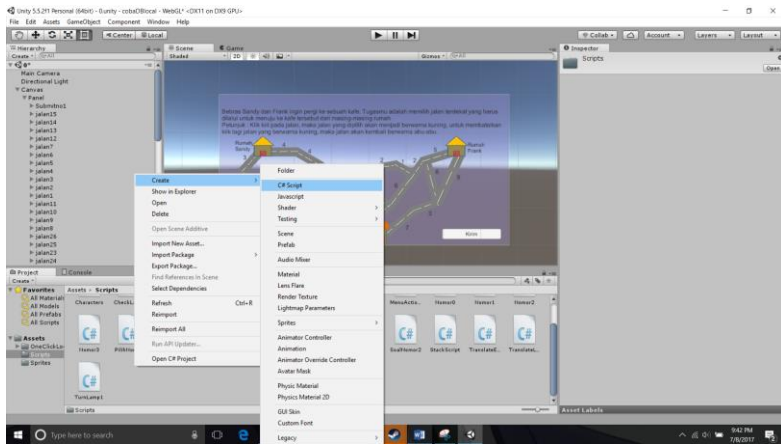
Gambar B.13 Membuat Teks



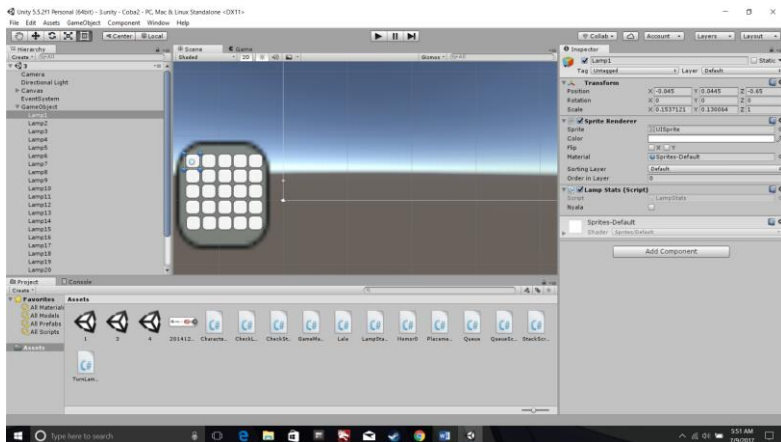
Gambar B.14 Membuat Tombol

1. Klik kanan pada *hierarchy* untuk membuat *canvas*, *canvas* ini diperlukan untuk menjadi latar dari penjelasan soal dan tombol seperti pada Gambar B.10.
2. Pilih warna yang diinginkan dengan cara memilih baris “Color” pada *canvas* seperti ditunjukkan pada Gambar B.11.
3. Setelah memilih warna, klik tombol “x” berwarna merah untuk selesai seperti pada gambar B.12.
4. Untuk membuat teks untuk penjelasan soal, maka klik kanan pada *canvas* lalu pilih menu “UI” dan pilih “Text” seperti pada gambar B.13.
5. Untuk membuat tombol aksi, maka klik kanan pada *canvas* lalu pilih menu “UI” dan pilih “Button” seperti pada Gambar B.14.

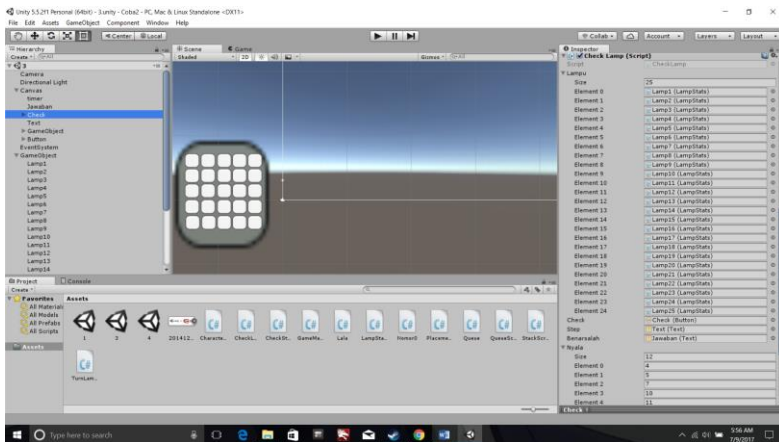
- Membuat Script



Gambar B.15 Membuat Script



Gambar B.16 Drag Script menuju Object



Gambar B.17 *Reference Object pada Script*

1. Klik kanan pada bagian *Project* di bawah, klik kanan lalu pilih “Create”, lalu pilih *C# Script* seperti pada Gambar B.15.
2. Untuk mengimplementasikan *Script*, pembuat soal harus *drag* menuju *Object* yang memerlukan *script* tersebut. Sebagai contoh, penulis mengimplementasikan *script LampStats* pada setiap *Game Object* lampu seperti pada gambar B.16.
3. Jika pada *Script* diharuskan melakukan *reference* pada objek tertentu maka akan muncul baris-baris seperti pada Gambar B.17. Sebagai contoh, penulis menginisialisasi *array* berisikan 25 lampu untuk *Script CheckLamp*. Maka pembuat soal harus *drag Game Object* lampu pada *hierarchy* menuju pada baris-baris pada *Script CheckLamp*.

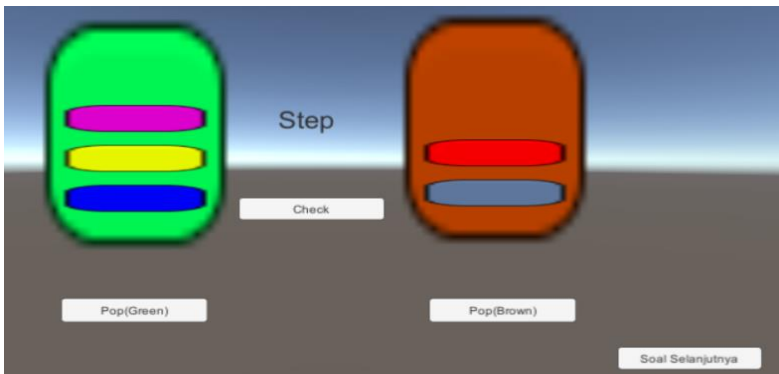
LAMPIRAN C

PANDUAN MENJAWAB SOAL

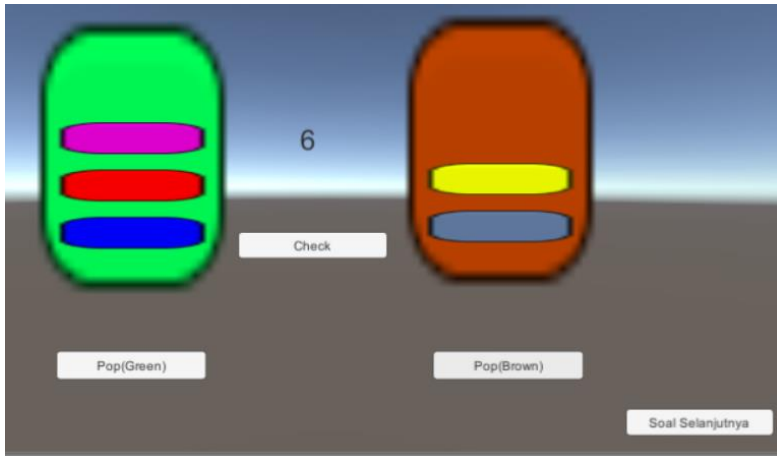
Berikut ini adalah lampiran untuk menjawab soal dengan benar pada aplikasi. Terdapat dua soal yang bermateri *Combinatorial*.

- Menjawab Soal Pertama

Soal pertama mengimplementasikan *stack* dan *queue* dimana terdapat dua wadah, berwarna hijau dan coklat. Terdapat dua tombol untuk operasi *pop* pada wadah berwarna hijau dan *dequeue* pada wadah berwarna coklat. Jika pengguna memilih tombol “Pop(Green)” maka piringan yang berada paling atas pada wadah berwarna hijau akan pindah menuju wadah berwarna coklat pada posisi paling atas. Sebaliknya, jika pengguna memilih tombol “Pop(Brown)” maka piringan yang berada paling bawah pada wadah berwarna coklat akan pindah menuju wadah berwarna hijau pada posisi paling atas. *State* awal ditunjukkan pada Gambar C.1 dan *state* akhir ditunjukkan pada Gambar C.2.



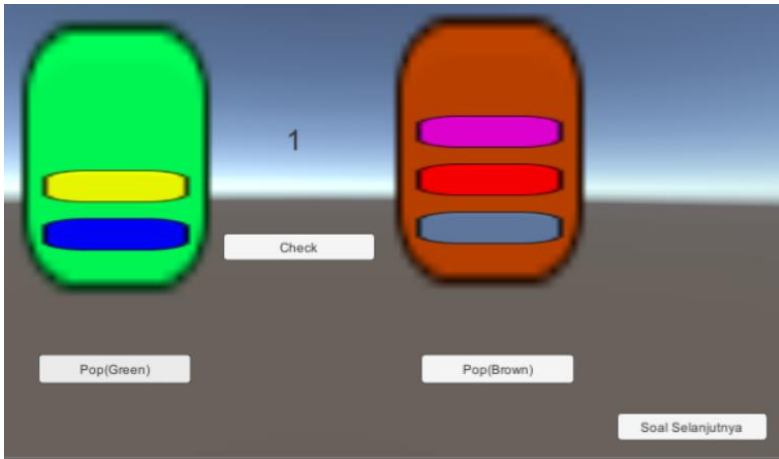
Gambar C.1 State Awal



Gambar C.2 State Akhir

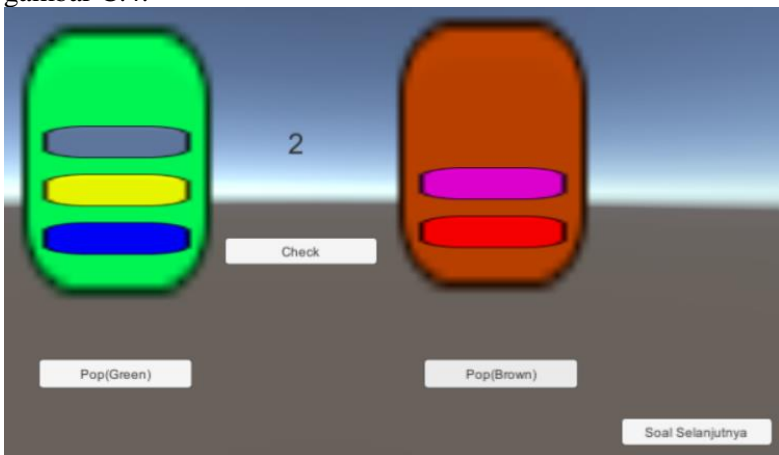
Seperti yang terlihat pada gambar C.2 bahwa untuk menuju *state* akhir membutuhkan 6 langkah. Dengan menggunakan dua prinsip *stack* dan *queue* yaitu FIFO dan LIFO, ada beberapa hal yang dapat membantu memecahkan masalah. Wadah hijau dapat digunakan untuk mengubah urutan piringan. Memindahkan dua piringan atau lebih dari wadah hijau menuju wadah coklat dapat mengubah urutan piringan. Wadah coklat dapat dijadikan tempat penyimpanan sementara untuk piringan-piringan. Karena piringan yang bergerak dari wadah coklat tidak akan mengubah tatanan piringan.

Pertama, pengguna memindahkan piringan berwarna ungu dari wadah berwarna hijau menuju wadah berwarna coklat seperti yang ditunjukkan pada gambar C.3



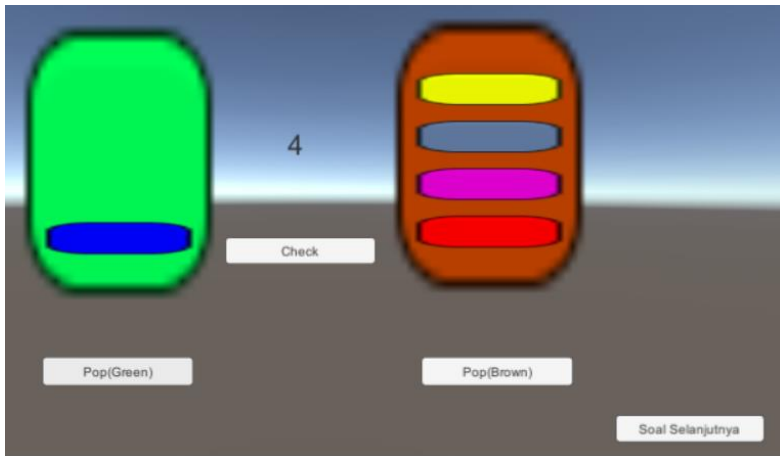
Gambar C.3 Langkah Pertama

Setelah piringan berwarna ungu berada pada wadah warna coklat, selanjutnya pengguna memindahkan piringan berwarna abu-abu menuju wadah berwarna hijau seperti pada gambar C.4.



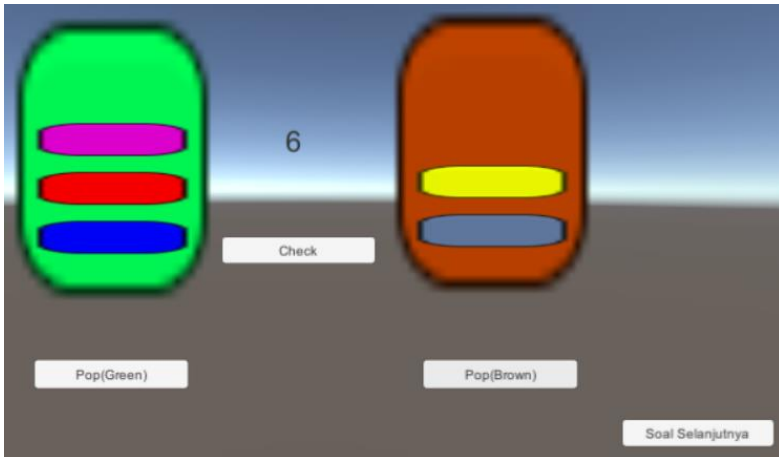
Gambar C.4 Langkah Kedua

Selanjutnya, pengguna memindahkan dua piringan paling atas pada wadah berwarna hijau menuju wadah berwarna coklat seperti ditunjukkan pada Gambar C.5.



Gambar C.5 Langkah Ketiga dan Keempat

Terakhir, pengguna memindahkan dua piringan paling bawah pada wadah berwarna coklat menuju wadah berwarna hijau seperti ditunjukkan pada Gambar C.6.



Gambar C.6 Langkah Kelima dan Keenam

Dari informasi diatas, maka dapat dirumuskan langkah minimum yang dibutuhkan untuk menukar piringan berwarna merah dan piringan berwarna kuning:

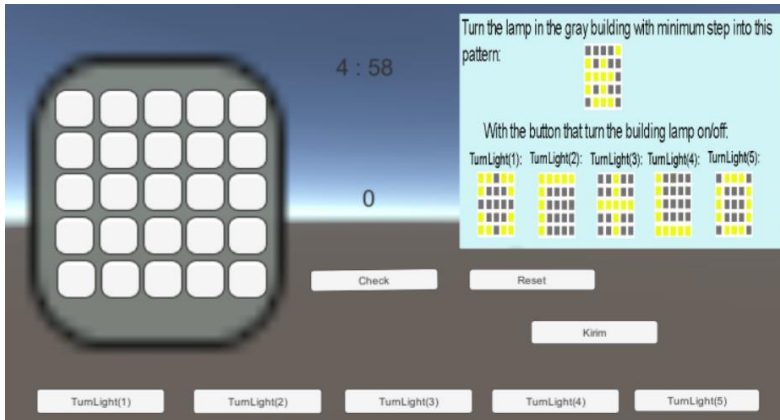
Pop(Green) → Pop(Brown) → Pop(Green) → Pop(Green) → Pop(Brown) → Pop(Brown).

Maka langkah minimum untuk menjawab soal pertama ini adalah enam langkah.

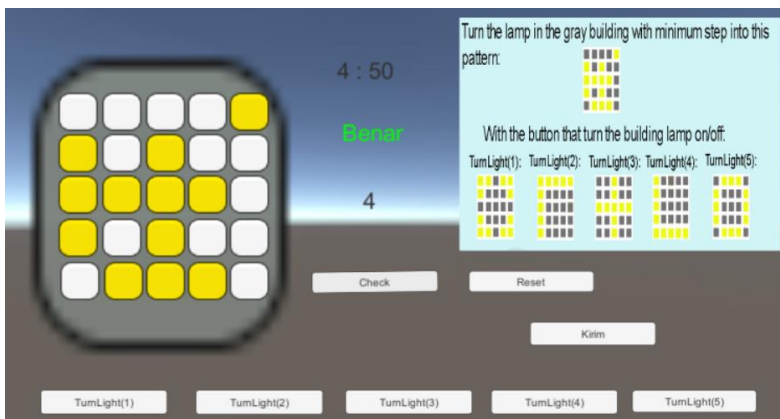
- **Menjawab Soal Kedua**

Soal kedua mengimplementasikan konsep bilangan biner, dimana bilangan biner terdiri dari dua *state* yaitu 0 dan 1, dan operasi XOR. Prinsip tersebut sama dengan soal kedua yang dibuat penulis dimana jika lampu menyala direpresentasikan dengan bilangan 1 dan lampu mati direpresentasikan dengan bilangan 0. Terdapat 25 lampu dengan state awal seperti

ditunjukkan pada Gambar C.7. Lampu-lampu tersebut harus dinyalakan seperti pada Gambar C.8.



Gambar C.7 State Awal Soal Kedua



Gambar C.8 State Akhir Soal Kedua

Seperti yang terlihat pada Gambar C.8 bahwa untuk dapat mencapai *state* akhir dibutuhkan empat langkah. Dengan

merepresentasikan lampu-lampu tersebut sebagai sebuah matriks maka akan didapatkan *state* awal dan *state* akhir sebagai berikut:

Initial

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Final

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 \end{bmatrix}$$


Untuk menemukan langkah minimum yang mengubah lampu dari *state* awal menuju *state* akhir adalah sebagai berikut:

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix} + \begin{bmatrix} 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 \end{bmatrix} + \begin{bmatrix} 1 & 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 \end{bmatrix}$$

$$\text{Initial} + \text{TurnLight}(3) + \text{TurnLight}(4) + \text{TurnLight}(5) + \text{TurnLight}(1) \\ = \text{Final}$$

Sehingga, empat langkah dibutuhkan untuk mencapai *state* kedua.

LAMPIRAN D KUISIONER



ITS
Institut Teknologi
Sepuluh Nopember

KUISIONER TUGAS AKHIR – 5113100121 CAYZA ANGKA MAULANA

**RANCANG BANGUN APLIKASI EDUKASI INTERAKTIF UNTUK MELATIH BERPIKIR KOMPUTASIONAL
MATERI KOMBINATORIAL**

Identitas Responden

Nama Lengkap : Putri Khumairatus Zahra Usia : 17 tahun

Pekerjaan : Pelajar Jenis Kelamin : L/P

A. PENILAIAN TERHADAP APLIKASI

Isilah tabel di bawah dengan menggunakan tanda (v)

SS = Sangat Setuju S = Setuju C = Cukup
TS = Tidak Setuju STS = Sangat Tidak Setuju

No	Parameter Antarmuka	STS	TS	C	S	SS
1	Aplikasi memiliki tampilan dan desain yang menarik				✓	
2	Aplikasi memiliki menu yang mudah digunakan				✓	
3	Aplikasi memiliki tata letak tombol yang sesuai				✓	
	Parameter Performa dan Kenyamanan					
4	Aplikasi nyaman untuk digunakan				✓	
5	Aplikasi diakses dengan cepat					✓
	Parameter Materi Soal					
5	Aplikasi memberikan soal yang menarik dan menantang					✓
6	Aplikasi memberikan soal yang sesuai dengan materi kombinatorial				✓	


B. KRITIK DAN SARAN

Pengguna sedikit bingung ketika membaca materi soal dan cara mengerjakan sehingga harus dipelajari lebih lanjut. Selain itu ditambahkan pembahasan materi yang lebih detail sebelum contoh soal agar pengguna bisa dengan mudah memahami. Ada beberapa soal yang kurang dipahami contohnya (kalkulasi dan grafik).

Surabaya, 30 Juni 2017

[Signature]

Gambar D.1 Kuisioner Responden Pertama



ITS
Institut Teknologi
Sepuluh Nopember

KUISIONER TUGAS AKHIR – 5113100121 CAYZA ANGKA MAULANA

**RANCANG BANGUN APLIKASI EDUKASI INTERAKTIF UNTUK MELATIH BERPIKIR KOMPUTASIONAL
MATERI KOMBINATORIAL**

Identitas Responden

Nama Lengkap : M. Hamzanudin Zuhdi Usia : 15 H

Pekerjaan : pelajar Jenis Kelamin : ♂ P

A. PENILAIAN TERHADAP APLIKASI

Isilah tabel dibawah dengan menggunakan tanda (v)

SS = Sangat Setuju S = Setuju C = Cukup

TS = Tidak Setuju STS = Sangat Tidak Setuju

No	Parameter Antarmuka	STS	TS	C	S	SS
1	Aplikasi memiliki tampilan dan desain yang menarik				✓	
2	Aplikasi memiliki menu yang mudah digunakan				✓	
3	Aplikasi memiliki tata letak tombol yang sesuai			✓		
	Parameter Performa dan Kenyamanan					
4	Aplikasi nyaman untuk digunakan				✓	
5	Aplikasi di akses dengan cepat			✓		
	Parameter Materi Soal					
5	Aplikasi memberikan soal yang menarik dan menantang					✓
6	Aplikasi memberikan soal yang sesuai dengan materi kombinatorial					✓

B. KRITIK DAN SARAN

pengecekan pd penilaian itu lebih baik diperbaiki
dan kalau soal lebih banyak lagi

Surabaya, 30 Juli 2017

[Signature]
M. Hamzanudin Z.

Gambar D.2 Kuisiонер Responden Kedua



KUISIONER TUGAS AKHIR – 5113100121 CAYZA ANGKA MAULANA

**RANCANG BANGUN APLIKASI EDUKASI INTERAKTIF UNTUK MELATIH BERPIKIR KOMPUTASIONAL
MATERI KOMBINATORIAL**

Identitas Responden

Nama Lengkap : Reviana Fadhilla Usia : 15 tahun
Pekerjaan : Pelayar Jenis Kelamin : t/P

A. PENILAIAN TERHADAP APLIKASI

Isilah tabel dibawah dengan menggunakan tanda (v)

SS = Sangat Setuju

S = Setuju

C = Cukup

TS = Tidak Setuju

STS = Sangat Tidak Setuju

No	Parameter Antarmuka	STS	TS	C	S	SS
1	Aplikasi memiliki tampilan desain yang menarik				✓	
2	Aplikasi memiliki menu yang mudah digunakan					✓
3	Aplikasi memiliki tata letak tombol yang sesuai				✓	
	Parameter Performa dan Kenyamanan					
4	Aplikasi nyaman untuk digunakan				✓	
5	Aplikasi di akses dengan cepat					✓
	Parameter Materi Soal					
5	Aplikasi memberikan soal yang menarik dan menantang					✓
6	Aplikasi memberikan soal yang sesuai dengan materi kombinatorial				✓	

B. KRITIK DAN SARAN

Soal terlalu panjang dan agak susah dipahami
Gambar lebih banyak menarik selengkapnya menantang

Surabaya, 27 Juni 2017

Reviana Fadhilla

Gambar D.3 Kuisisioner Responden Ketiga

BIODATA PENULIS



Penulis lahir di kota Surabaya pada tanggal 13 Desember 1995, merupakan anak bungsu dari 3 orang bersaudara. Dalam perjalanan hidupnya penulis pernah menempuh pendidikan dasar di SD Negeri Klampis Ngasem I/246 Surabaya, dilanjutkan pendidikan menengah SMP Negeri 30 Surabaya, tingkat atas di SMA Negeri 17 Surabaya dan terakhir di S1 Departemen Teknik Informatika Institut Teknologi Sepuluh Nopember (ITS Surabaya) di rumpun mata kuliah Algoritma dan Pemrograman (AP).

Selama menjadi mahasiswa penulis pernah mengikuti pelatihan pengembangan diri mahasiswa dari tingkat Pra-Dasar, Dasar, Schematics ITS 2014 dan 2015. Untuk bidang Organisasi penulis pernah tergabung di HMTC ITS ditahun 2014 sampai 2016. Penulis dapat dihubungi lewat email cayzamaulana@gmail.com.